

参赛密码 _____
(由组委会填写)



第十二届“中关村青联杯”全国研究生 数学建模竞赛

题目

旅游路线优化设计问题研究

摘 要

本文分别通过建立带约束的非线性单目标和多目标优化模型，解决了旅游路线的最优设计问题。线路的设计主要受旅游费用、旅游时间、可游览景点数等因素制约，通过将路线规划问题转化为旅行商（TSP）问题，使用蚁群算法和遗传算法求解得到最优的旅游路线。

问题一要求设计最优的旅游路线在最少的年数内游完全国 201 个 5A 景点。本文建立带约束的非线性单目标优化模型，以旅游时间花销最少为优化目标。将普通公路里程和高速公路里程统一转化为高速公路里程，推导出总旅游时间花费函数表达式。合理地引入 0-1 变量表示旅行者是否游玩过景点，得到将旅游时间转化为天数的分段函数表达式。将旅游路途时间花销和游览时间花销转化为景点 i 到景点 j 的旅游天数时间花销，从而将旅游路线规划问题转化为 TSP 问题，进而通过编程使用蚁群算法求解得到最优的旅游路线。获得最优旅游路线及时间花销之后，结合人工判断合理的旅游花销天数（即满足一年旅游时间不超过 30 天，每年外出次数不超过 4 次）组合成一年的总时间花销进而得到旅游的最少总年数。

问题二要求在十年内游完 201 个 5A 级景区，并综合考虑多种出行方式，给出费用最优、费用最优、旅游体验最好的旅游线路。本文建立了带约束的非线性多目标约束模型，以费用最优和体验最佳为优化目标。与问题一不同的是，本问要求花费时间最少、体验最佳，本文对体验最佳的理解为交通耗时最少而游览时间较多。同样地，本题也可以用问题一的方法，将旅游路途交通花费和游览时间花销转化为景点 i 到景点 j 的旅游天数费用花销，将问题二转化为 TSP 问题。使用遗传算法求解得到花销最少的旅游路线集，结合人工判断从中挑选出时间花销最短的旅游路线。

问题三需要在第二问所建立的模型基础上加以推广，为全国的自驾游爱好者规划设计旅游线路并给旅游爱好者和旅游有关部门提出建议。为使模型更具通用性，本文将旅行者的餐饮费和不同景点的门票费用加入到拓展的模型之中，并推导出旅游总花费函数表达式和总游览时间函数表达式。和问题二类似，将旅游路途交通花费和游览时间花销转化为景点 i 到景点 j 的旅游天数费用花销，将问题三转化为 TSP 问题，通过编程使用遗传算法求解得到花销最少的旅游路线集，结合人工判断从中挑选出时间花销最短的旅游路线。结合前三问的结果，并进行分析给旅游相关部门提出了两条建议：（1）完善景区之间的高速公路建设，（2）在热门景点设置一定数量的停车场；给旅游者提出了三条建议：（1）尽量延长每次出游时间，（2）可以考虑组团自驾游，（3）避开旅游热季。

问题四要求参考附件 6 与附件 7，更为合理地规划未来十年的旅游计划。通过改进模糊层次分析法模型，从可能被摘牌的 5A 景区所在省份选拔出评分较高的 4A 景区作为备选 5A 景区。本文将 5A 景区的评定细则分为服务质量与环境质量、景观质量、游客意见三部分作为模型的准则层，每个部分的评定款项作为方案层。由细则给定的打分作为各款项的权重。由改进模糊层次分析法模型得到各省内 4A 景区的综合排名，从而得到最优的备选 4A 景区。以最优的 4A 景区替换可能被摘牌的 5A 景区，使用问题三

的算法进行求解，得到更为合理地规划未来十年的旅游计划。

基于本文较为完善的机理分析，建立带约束的非线性单目标和多目标优化模型，并借助蚁群算法和遗传算法结合的方式，得到符合要求的最优路线设计方案。成功地对 0-1 变量进行使用和约束，简化模型建立难度，得到较好的结果。最后本文对模型进行评价与推广，模型的精准性高、鲁棒性强、普适性强。

关键词：旅行路线规划 多目标优化 蚁群算法 遗传算法 TSP 问题 0-1 变量

一、问题重述及国内外研究情况

1.1 问题的背景

旅游活动正在成为全球经济发展的重要动力之一，它加速国际资金流转和信息、技术管理的传播，创造高效率消费行为模式、需求和价值等。随着我国国民经济的快速发展，人们生活水平得到很大提升，越来越多的人积极参与有益于身心健康的旅游活动。

1.2 问题的提出

附件给出了全国 201 个 5A 级景区的名单，全国高速公路，全国火车、高铁、飞机班次等信息。一位自驾游爱好者拟按这些附件制定旅游计划。根据该旅游爱好者的个人偏好，景点位置及开放时间的实际情况，在旅行中需要达到以下条件：

该旅游爱好者每年有不超过 30 天的外出旅游时间，每年外出旅游的次数不超过 4 次，每次旅游的时间不超过 15 天；

- 根据个人偏好，每个 5A 级景区的游览时间不得小于附件中的要求，最长逗留时间不得超过附件中最少时间的 2 倍；

- 基于安全考虑，行车时间限定于每天 7:00 至 19:00 之间，每天开车时间不超过 8 小时；

- 若是全天游览，则开车时间控制在 3 小时内；若是半天游览，开车时间控制在 5 小时内；

- 在高速公路上的行车平均速度为 90 公里/小时，在普通公路上的行车平均速度为 40 公里/小时；

- 该旅游爱好者计划在每一个省会城市至少停留 24 小时，以安排专门时间去游览城市特色建筑和体验当地风土人情（不安排景区浏览）；

- 选择高铁出行要求当天乘坐高铁的时间不超过 6 个小时，乘坐高铁或飞机的当天至多安排半天的景区游览；

- 景区开放时间统一为 8:00 至 18:00；

- 旅行中租车费用 300 元/天，油费和高速过路费另计，租车和还车需在同一城市；

- 住宿费简化为省会城市和旅游景区 200 元/人·天，地级市 150 元/人·天，县城 100 元/人·天；高速公路的油耗加过路费平均为 1.00 元/公里，普通公路上油耗平均为 0.60 元/公里；

根据上述条件，需要解决下面问题：

(1) 该旅行者出行先通过高速公路到达与景区邻近的城市，再自驾到景区。以其常住地在西安市为例，规划设计旅游线路，试确定游遍 201 个 5A 级景区至少需要几年？给出每一次旅游的具体行程（每一天的出发地、行车时间、行车里程、游览景区）。

(2) 若出行方式考虑乘坐高铁或飞机到达与景区相邻的省会城市，而后租车自驾到景区游览。根据附件材料，建立数学模型设计一个十年游遍所有 201 个 5A 景区、费用最优、旅游体验最好的旅游线路，给出每一次旅游的具体线路（含每次具体出行方式；每一天的出发地、费用、路途时间、游览景区、每个景区的游览时间）。

(3) 在 (2) 的基础上加以推广，为全国的自驾游爱好者规划设计类似的旅游线路，进而给出常住地在北京市的自驾游爱好者的十年旅游计划；根据上述三问的结果给旅游爱好者和旅游有关部门提出建议。

(4) 根据国家 5A 级旅游景区评定的相关信息，更合理地规划该旅游爱好者的十年旅游计划。

1.3 国内外现状

国内外有学者对于旅游路线规划等问题也进行了相关研究,佟欣^[1]等人将黑龙江 30 个景点之间的关系转化为图论问题,建立赋权图,利用蚁群算法来解决最短路径问题,从而研究出黑龙江的最优旅游路线。潘玉侠^[2]等人根据旅游线路设计的特点对遗传算法进行了改进,建立了基于遗传算法的旅游线路优化算法。国外部分学者^[3-5]等也针对旅游路线规划等相关问题进行了研究,得到了较为丰富的结果。

二、基本假设

1. 假设未来十年多的物价等维持在现在水平;
2. 途中没有遇到意外情况,行程没有延误;
3. 计算交通耗时四舍五入像一刻钟取整;
4. 除了当天能够游玩两个景点的情况,旅行者在到达景点后不在安排行车;
5. 票价以当前时间为准,未发生变动;
6. 假设被警告的 9 个 5A 景区在未来被摘牌;
7. 升级的 4A 景区从被摘牌的 9 个 5A 景区所在省中选取。

三、符号说明与名词解释

3.1 符号说明

序号	符号	符号说明
1	t_i	在景点 j 停留时间
2	$m(k)$	第 k 次出游游玩的景点数
3	r_{ij}	是否从景点 i 到景点 j
4	$T_{ij}^{(k)}$	第 k 次出游为从第 i 个景点到景点 j 路上花费的时间和景点 j 游览时间转化后的天数
5	$T_d^{(k)}$	第 k 次旅游游览时间为两天的次数
6	$T^{(k)}$	第 k 次旅游游览的天数
7	d_{ij}	两个城市 i 与 j 之间的高速公路距离
8	C_T	交通费总和
9	C_D	住宿费总和
10	C_R	餐饮费总和
11	C_i	为景点 i 的门票费

3.2 名词解释

序号	名词	名词解释
1	国家 5A 级旅	中华人民共和国旅游景区质量等级划分为五级,从高到低依次为

	游景区	AAAAA、AAAA、AAA、AA、A 级。5A 级为中国旅游景区最高等级，代表着中国世界级精品的旅游风景区等级。
--	-----	---

四、问题分析

4.1 问题一的分析

本问要求一种设计合适的方法，建立数学模型，以旅游爱好者的常住地在西安市为例，规划设计旅游线路，试确定游遍 201 个 5A 级景区至少需要几年。该问题与 TSP 问题类似，但是又不是常见的 TSP 问题，因此考虑通过相应的转换，将该问题转化为 TSP 问题。该问的约束较多，需要逐一满足，同时要满足总体约束即满足一年旅游时间不超过 30 天，每年外出次数不超过 4 次。通过查询各景点之间的公路里程，我们将普通公路和高速公路混合的路线通过加权的方式统一转化为高速公路以方便计算。将旅游路途时间花销和游览时间花销转化为景点 i 到景点 j 的旅游天数时间花销，从而将旅游路线规划问题转化为 TSP 问题，这类问题可以通过二叉树、蚁群算法、遗传算法等算法来求解。

4.2 问题二的分析

问题二要求在十年内游完 201 个 5A 级景区，并综合考虑出行方式，给出费用最优的费用最优、旅游体验最好的旅游线路。这是一个带约束的非线性多目标约束问题，以费用最优和体验最佳为优化目标。与问题一不同的是，本问要求花费时间最少、体验最佳，我们对体验最佳的理解为交通耗时最少而浏览时间较多。与第一问类似，本问可以同样地将旅游路途交通花费和游览时间花销转化为景点 i 到景点 j 的旅游天数费用花销，将问题二转化为 TSP 问题。在求解时，可以通过使用遗传算法和蚁群算法结合的方式，通过遗传算法播撒可能的结果种子，蚁群算法计算花费时间。当得到几组费用花销最少的旅游线路之后，可以从中筛选出时间花销最少的旅游线路。

4.3 问题三的分析

问题三需要在问题二所建立的模型基础上加以推广，为全国的自驾游爱好者规划设计旅游线路并给旅游爱好者和旅游有关部门提出建议。为使模型更具通用性，我们可以考虑将旅行者的餐饮费和不同景点的门票费作加入到问题二的模型之中。模型建立之后，可以采用与问题二类似的算法求解。最后需要结合前三问的结果进行分析给旅游相关部门和旅游者提出建议。

4.4 问题四的分析

问题四要求参考附件 6 与附件 7，更为合理地规划未来十年的旅游计划。通过附件 6 可以知道 2015 年 4 月 2 日上午，国家旅游局通报，从 2014 去年四季度到 2015 年一季度，山西忻州五台山、南京夫子庙秦淮河观光带等 9 家 5A 景区被警告。我们考虑将可能被摘牌的 5A 景区所在省份选拔出评分较高的 4A 景区作为备选 5A 景区。这样可以更好地规划未来十年的旅游路线。因为在附件 6 中给出了 5A 级景区的评定细则，考虑建立层次分析法来合理地筛选备选 5A 景区。将备选 5A 替换原可能被摘牌的 5A 景区之后，收集替换后的 5A 景区各景点的公路里程，接下来可以重用问题三的算法进行求解，并得到最优路线。

五、问题一模型的建立、求解及应用

本文要求一种设计合适的方法，建立数学模型，以旅游爱好者的常住地在西安市为例，规划设计旅游线路，试确定游遍 201 个 5A 级景区至少需要几年。本文在百度地图^[6]上收集数据，该数据包括两部分：省内景点与景点之间的最短距离（其中包括高速公路距离和普通公路）和各省会与景点之间的距离。景点大致分布图如下图 1 所示。在满足相应的约束条件下，建立数学模型计算得出最佳路线，从而得到时间（天数）最少的目标函数，并对目标函数里的参数进行多目标约束，采用蚁群算法求解出以西安为起点的旅游路线，并得到每一次旅游的具体行程（每一天的出发地、行车时间、行车里程、游览景区等），从而可得出至少需要的时间，解题思路如图 2 所示。



图 1 A5 景点大致分布图

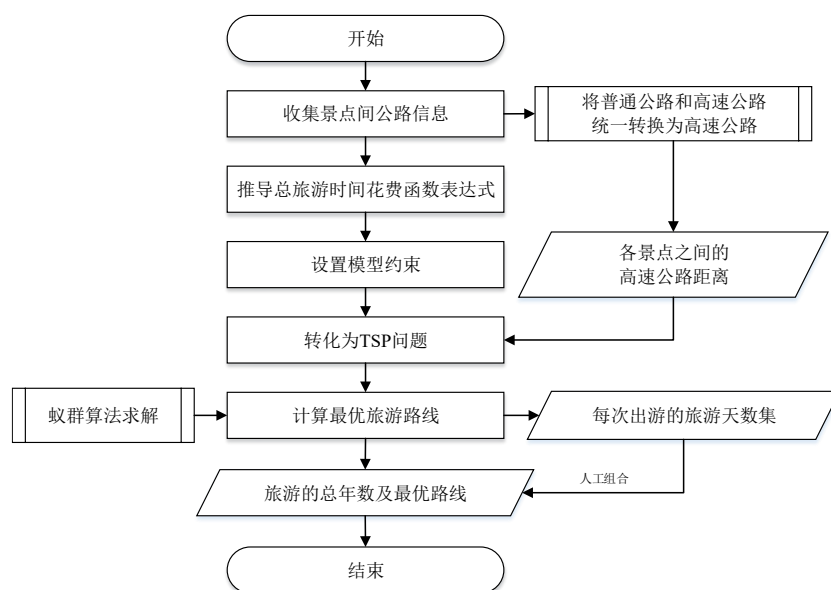


图 2 问题 1 的思路流程图

5.1 模型的建立

5.1.1 目标函数的确立

经分析,此问的目标是用最短的旅游时间和次数游览完 201 个景点,本文首先查出这 202 个景点(包括出发地以及 201 个旅游景点),每个地点到其他 201 个地点最快捷的交通工具及其费用。在满足相应的约束条件下,建立数学模型计算得出最佳路线。

游览的总时间包括在各个景点的停留时间、两个景点间交通所花的时间,由此的目标函数:

$$\min T = T_1 + T_2 \quad (1)$$

(1) 总停留时间

t_i 表示停留时间,即游客在 i 景点游览时间; t_j 表示停留时间,即游客在 j 景点游览时间; $r_{ij}^{(k)}$ 表示是否从 i 景点到 j 景点,而整个旅游线路是一个环形,因而

$\sum_{i=1}^{202} \sum_{j=1}^{202} r_{ij} (t_i + t_j)$ 实际上表示游客在所到景点的停留时间算了两遍,所以总停留时间为:

$$T_1 = \frac{1}{2} \sum_{i=1}^{202} \sum_{j=1}^{202} r_{ij} (t_i + t_j) + \sum_{i=1}^{202} w_i \times u_i \quad (2)$$

(2) 总交通时间

t_{ij} 表示从 i 地到 j 地所花费的时间, d_{ij} 为 i 景点到 j 景点普通公路的距离, d_{ij}^p 是从 i 景点到 j 景点普通公路的距离, d_{ij}^s 是从 i 景点到 j 景点高速公路的距离。由此可得总交通时间为:

总距离为:

$$d_{ij} = d_{ij}^p + d_{ij}^s$$

由题意可知在高速公路上的行车平均速度为 90 公里/小时,在普通公路上的行车平均速度为 40 公里/小时。所以,所花的时间为:

$$t_{ij} = d_{ij}^p / 40 + d_{ij}^s / 90$$

总交通时间:

$$T_2 = \sum_{i=1}^{202} \sum_{j=1}^{202} t_{ij} r_{ij} = \sum_{i=1}^{202} \sum_{j=1}^{202} (d_{ij}^p / 40 + d_{ij}^s / 90) r_{ij} \quad (3)$$

综上可得,目标函数为:

$$\min T = T_1 + T_2 = \frac{1}{2} \sum_{i=1}^{202} \sum_{j=1}^{202} r_{ij} (t_i + t_j) + \sum_{i=1}^{202} w_i \times u_i + \sum_{i=1}^{202} \sum_{j=1}^{202} (d_{ij}^p / 40 + d_{ij}^s / 90) r_{ij} \quad (4)$$

5.1.2 约束条件

(1) 时间约束

根据假设景点的开放时间为 8:00-18:00,所以在每个景点可停留的最多时间为 10 个小时,可得约束条件为:

$$t_i \leq 10 \quad (i=1,2,3,\dots,202)$$

该旅游爱好者计划在每一个省会城市至少停留 24 小时，所以约束条件为：

$$t_s \geq 24 \quad (i=1,2,3,\dots,202)$$

(2) 旅游景点数约束

根据假设，这个旅游路线是环形，即最终游客要回到西安，因此 R 表示游客旅游的景点数，游客要游玩 202 个景点， $m(k)$ 是第 k 次出游所游玩的景点数量。即

$$\sum_{i=1}^{m(k)} \sum_{j=1}^{m(k)} r_{ij} = m(k)$$

(3) 0—1 变量约束

由于旅游路线是环形的，把各个景点连成一个圈，把每个景点看成圈上的一个点。对于每个点来说，只允许最多一条边进入，同样只有一条边出来，并且当有一条边进入时必有一条边出来。由此可得约束条件：

$$\sum_{i=1}^{m(k)} r_{ij} = \sum_{j=1}^{m(k)} r_{ij} \leq 1 \quad (i, j = 1, 2, \dots, 202)$$

当 $i=0$ 时，因为西安是出发点，所以 $\sum_{i=1} r_{ij} = 1$ ；

当 $j=1$ 时，因为游客最终回到西安，所以 $\sum_{j=1} r_{ij} = 1$ 。

综上所述，

$$\sum_i r_{ij} = \sum_j r_{ij} \leq 1 \quad (i, j = 1, 2, \dots, 202)$$

$$\sum_{i=1} r_{ij} = 1 \text{ 和 } \sum_{j=1} r_{ij} = 1$$

同样，当 $i, j \geq 2$ 时，根据题意不可能出现 $r_{ij} = r_{ji} = 1$ ，即不可能出现游客往返旅游，由此可得约束条件：

$$r_{ij} \times r_{ji} = 0$$

(4) 停留时间约束

由于游客在各个景点都有一个最短停留时间，可得约束条件：

$$t_i > 0$$

(5) 时间转化（将游览时间和路程时间转化为天数）

由题目可知当安排全天游览则开车时间控制在 3 小时内，安排半天景点游览，开车时间控制在 5 小时内。可以得到：

$$t_j = \begin{cases} 3 \text{小时} & \text{游览半天} \\ 5 \text{小时} & \text{游览一天} \end{cases}$$

这里假设旅行者只在早晨开始型车，旅行者到达景点后，不再安排当天的行车且市内行车不计行车安排，即 $t_{ij} = 0$ 。 t_{ij} 为从第 i 个景点到第 j 景点路上花费的时间和第 j 景

点游览时间转化后的天数。即可以得出：

$$T_{ij}^{(k)} = \begin{cases} 1 & t_{ij} = 0; \quad t_j = 3 \text{ 且 } t_{ij} > 3 \\ & 1 \quad t_{ij} = 0; \quad t_j = 5 \\ 1 & t_{ij} = 0; \quad t_j = 3 \text{ 且 } t_{ij} = 3 \\ & 1 \quad t_{ij} \leq 5; \quad t_j = 3 \\ & 1 \quad t_{ij} \leq 3; \quad t_j = 5 \\ 2 & 8 < t_{ij} \leq 13; \quad t_j = 3 \\ 2 & 8 < t_{ij} \leq 11; \quad t_j = 5 \\ 3 & 16 < t_{ij} \leq 21; \quad t_j = 3 \\ 3 & 16 < t_{ij} \leq 19; \quad t_j = 5 \end{cases}$$

$T_d^{(k)}$ 为第 k 次旅游，游览时间为两天的次数。则第 k 次旅游的总天数为：

$$T^{(k)} = T_{ij}^{(k)} + T_d^{(k)}$$

每次旅游的时间不超过 15 天，即满足下面的公式：

$$T^{(k)} \leq 15 \quad \text{且} \quad \sum m^{(k)} = 201$$

约束条件为：

$$\min \sum_{k=1}^n T^{(k)} \leq 30 \quad (k \leq n \leq 4 \text{ 次})$$

5.1.3 模型建立

综上所述，可得总模型为：

$$\min T = T_1 + T_2 = \frac{1}{2} \sum_{i=1}^{202} \sum_{j=1}^{202} r_{ij} (t_i + t_j) + \sum_{i=1}^{202} w_i \times u_i + \sum_{i=1}^{202} \sum_{j=1}^{202} t_{ij} r_{ij} \quad (5)$$

约束条件为：

$$\left\{ \begin{array}{l} \sum_{i=1}^{202} \sum_{j=1}^{202} r_{ij} = 202 \\ \sum_i r_{ij} = \sum_j r_{ij} \leq 1 (i, j = 1, 2, \dots, 11) \\ \sum_{i=1} r_{ij} = 1, \sum_{j=1} r_{ij} = 1 \\ r_{ij} \times r_{ji} = 0 \\ t_i > 0 \\ T^{(k)} \leq 15 \\ \sum m^{(k)} = 201 \\ \min \sum_{k=1}^n T^{(k)} \leq 30 (k \leq n \leq 4) \end{array} \right. \quad (6)$$

5.2 模型求解及分析

将各个旅游景点之间的关系转化为图论问题，并做以下分析：

建立有向图 $G = (V, A)$ 。其中 $V = \{V_1, V_2, \dots, V_n\}$ 称为图 G 的顶点集， V 中的每一个

元素 $V_i (i=1,2,\dots,n)$ 称为该图的一个顶点，在该题中表示 n 城市； $A = \{a_1, a_2, \dots, a_n\}$ 称为图 G 的弧集， A 中的每个元素 $a_k = (V_i, V_j)$ 称为该图的一条从 V_i 到 V_j 的弧，在此题中表示各个城市两两连线的集合。^[1]

设城市个数为 n ， d_{ij} 表示两个城市 i 与 j 之间的距离， $x_{ij} = 0$ 或 1 (1 表示走过城市 i 到城市 j 的路， 0 表示没有选择走这条路)。本题可以向 TSP 问题进行转化，则 TSP 问题的数学模型为：

$$\min \sum_{i \neq j} d_{ij} x_{ij} \quad (7)$$

5.2.1 蚁群算法求解

(1) 状态转移规则

因为蚂蚁 k 不能重复经过一个城市，所以建立禁忌表 $tabu_k (k=1,2,\dots,m)$ 来记录蚂蚁走过的城市，禁忌表随着时间做动态变化。

建立蚂蚁 k 由 i 城市转移到 j 城市的状态转移概率如下：

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ik}(t)]^\beta}{\sum_{s \in tabu_k} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta} & j \notin tabu_k \\ 0 & j \in tabu_k \end{cases} \quad (8)$$

上式中 α 为信息启发式因子，表示路径的相对重要性，是对所积累的信息素影响作用的一个加权值； β 为期望启发式因子，表示能见度的相对重要性；

每只蚂蚁必须依据以城市距离和连接边上信息素的数量为变量的概率函数，决定选择下一个城市的概率。

每只蚂蚁必须根据禁忌表和概率函数寻找下一个城市，以保证该蚂蚁从起点出发经过所有城市有且只有一次，并且最终返回到起点。

(2) 信息素的全局更新规则

当 m 只蚂蚁成功的完成一次寻径过程之后，将选出目标函数值最小的路径，用以完成全局信息素的更新，使得较优解保留下来，对后继蚂蚁产生影响，加快收敛到最优解的速度。

设 i, j 为两个相连接点，则有：

$$\tau_{ij}(i, j) \leftarrow (1 - \rho) \tau_{ij}(i, j) + \rho \times \Delta \tau_{ij}(i, j) \quad (9)$$

其中，变量 $\Delta \tau_{ij}(i, j)$ 是在 t 时刻，节点 i, j 之间路上信息素的增加量

$$\Delta \tau_{ij}(i, j) = \begin{cases} (L_{\text{短}})^{-1} & \text{if } (i, j) \in \text{global - best - tour} \\ 0 & \text{otherwise} \end{cases}$$

ρ 是位于 $[0, 1]$ 上的“激素”挥发因子； $L_{\text{短}}$ 为到目前为止所找到全局最短路径长度。

(3) 信息素的局部更新

对于第 k 只蚂蚁，在建立一个解得过程中也同时进行激素迹的更新，如果节点 i, j 是它所选择路径上的两个相邻节点，规则如下：

$$\tau_{ij}(t) \leftarrow (1 - \rho) \tau_{ij}(t) + \rho \times \Delta \tau_{ij}(t)$$

否则，不更新。其中， $0 < \rho < 1$ ， $\Delta \tau_{ij}(t) = \tau_0$ ， τ_0 是各条路上的信息素的初始值，通常取同一值，表示同一环境。

信息素的更新策略有很多种方法，每种更新策略的主要差别体现在 $\Delta \tau_{ij}^k(t)$ 的求法

上。我们规定蚂蚁在完成一个循环后更新所有路径上的信息素，其方程式为：

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k} & \text{k蚂蚁本次循环经过 (i,j)} \\ 0 & \text{否则} \end{cases} \quad (10)$$

上式中 Q 表示蚂蚁携带信息素的量，其值的大小影响算法的收敛速度； L_k 表示第 k 只蚂蚁在本次循环中所走的路程总长度。

5.2.2 基于蚁群算法的实现步骤^[6]

本题基于蚁群算法的实现步骤如下：

- step1*：初始化。时间 $t = 0$ ，循环次数 $n_c = 0$ ，设置最大循环次数为 $n_{c_{\max}}$ ， $\Delta\tau_{ij}(0) = 0$ ；
- step2*：循环次数 $n_c ++$ ；
- step3*：蚂蚁个数 $k ++$ ；
- step4*：蚂蚁选择可以到达的城市，按照状态转移规则移动到下一个城市 j ；
- step5*：对于城市 j ，由于已经到达，所以添加到禁忌表中；
- step6*：判断所有城市是否都经过，若未完全经过，表明蚂蚁个数没有达到 m ，则转向执行 *step3*，否则执行 *step7*；
- step7*：由于信息素改变，要求按照公式 (2) (3) 更新最短路径信息素，使得较优解保留，加快收敛到最优解的速度；
- step8*：若 $n_c < n_{c_{\max}}$ 表明没有满足终止条件，即转向执行 *step2*，否则执行 *step9*；
- step9*：输出最优结果。

5.2.3 求解结果及分析

通过上文计算，得到该旅游爱好者的常住地在西安市为例，规划设计其最优的旅游线路，下表为游遍 201 个 5A 级景区的路线以及每一次旅游的具体行程（每一天的出发地、行车时间、行车里程、游览景区。表 1 和表 2 分别为第一次出行路线具体行程和该旅游爱好者游遍 201 个 5A 级景区的路线。

表 1 第一次出行路线具体行程（其他次出行详见附录）

时间	消耗时间 (天)	出发地	行车里程 (公里)	游览景区
第 1 天 上午	0.5	西安市	7.7	西安大雁塔—大唐芙蓉园景区
第 1 天 下午	0.5	西安大雁塔—大唐芙蓉园景区	44	西安秦始皇兵马俑博物馆
第 2 天 上午	0.5	西安秦始皇兵马俑博物馆	9.1	西安华清池景区
第 2 天 下午	0.5	西安华清池景区	151.7	宝鸡扶风县法门寺佛文化景区
第 3 天	1	宝鸡扶风县法门寺佛文化景区	227.4	渭南华阴市华山风景区
第 4 天	1	渭南华阴市华山风景区	282	延安黄陵县黄帝陵景区
第 5 天	1.2	西安市	880	西宁市
第 6 天	0.5	西宁市	25.5	西宁市湟中县塔尔寺景区

第 7 天	1	西宁市湟中县塔尔寺景区	27.5	青海湖风景区
第 8 天	0.9	西宁市	640	银川市
第 9 天 上午	0.5	银川市	25	银川市灵武水洞沟旅游区
第 9 天 下午	0.5	银川市灵武水洞沟旅游区	56.9	银川镇北堡西部影视城
第 10 天	0.5	银川镇北堡西部影视城	251.6	中卫沙坡头旅游景区
第 11 天	1	中卫沙坡头旅游景区	241.9	石嘴山平罗县沙湖旅游景区
第 12 天	0.9	银川市	670	呼和浩特市
第 13 天	1	呼和浩特市	184.2	鄂尔多斯达拉特旗响沙湾旅游景区
第 14 天	0.5	鄂尔多斯达拉特旗响沙湾旅游景区	112.4	鄂尔多斯伊金霍洛旗成吉思汗陵旅游区
第 14 天		鄂尔多斯伊金霍洛旗成吉思汗陵旅游区	270	无
第 15 天	1.3	呼和浩特市	970	西安市



图 3 第一次出行的总体路线

由题意要求，每年有不超过 30 天的外出旅游时间，每年外出旅游的次数不超过 4 次，每次旅游的时间不超过 15 天，则该旅游者完成 201 个 5A 级景区的总次数以及时间和路线如下表 2 所示。

表 2 出行的总体路线及时间

次数	路线	用时(天)	换算时间(年)	景点(个)
起点	省内景点	4 天	1 年	6
第一次路线	西安→西宁→银川→呼和浩特→西安	15 天		14

第二次路线	西安→天津→上海→西安	8.4 天		5
第三次路线	西安→拉萨→西安	8.8 天	1 年	2
第四次路线	西安→贵阳→海口→西安	11.8 天		9
第五次路线	西安→北京→长春→西安	11.7 天	1 年	11
第六次路线	西安→太原→沈阳→西安	12.4 天		10
第七次路线	西安→南宁→兰州→西安	14.9 天	1 年	8
第八次路线	西安→重庆→西安	8 天		5
第九次路线	西安→石家庄→西安	7.7 天	1 年	5
第十次路线	西安→长沙→西安	8.5 天		7
第十一次路线	西安→济南→西安	9 天		9
第十二次路线	西安→合肥→西安	10.1 天	1 年	8
第十三次路线	西安→昆明→西安	10.4 天		6
第十四次路线	西安→南昌→西安	11 天	1 年	7
第十五次路线	西安→郑州→西安	10.9 天		10
第十六次路线	西安→哈尔滨→西安	12.4 天	1 年	5
第十七次路线	西安→福州→西安	12.6 天		8
第十八次路线	西安→广州→西安	12.1 天	1 年	10
第十九次路线	西安→杭州→西安	12.1 天		12
第二十次路线	西安→成都→西安	11 天	1 年	10
第二十一次路线	西安→武汉→西安	12.8 天		11
第二十二次路线	西安→乌鲁木齐→西安	13.5 天	1 年	5
第二十三次路线	西安→乌鲁木齐→西安	13.5 天		4
第二十四次路线	西安→南京→西安	15 天	1 年	19
总计		265.6 天	12 年	201

由于时间的限制，在新疆自治区需要旅游两次，如下表 3 所示。

表 3 西安到新疆旅游的具体路线

消耗时间 (天)	出发地	行车里程(公里)	游览景区
3.5	西安市（第一次）	2540	乌鲁木齐市
0.5	乌鲁木齐市	189.1	吐鲁番葡萄沟风景区
1	吐鲁番葡萄沟风景区	207.1	乌鲁木齐天山大峡谷
1	乌鲁木齐天山大峡谷	145.7	昌吉州阜康市天山天池风景名胜 区
1.5	昌吉州阜康市天山天池风 景名胜区	409	巴音郭楞蒙古自治州博湖县博 斯腾湖景区
1.5	巴音郭楞蒙古自治州博湖 县波斯腾湖景区	356.7	伊犁地区新源县那拉提旅游风 景区
3.5	乌鲁木齐市	2540	西安市
3.5	西安市（第二次）	2540	乌鲁木齐市
1.5	伊犁地区新源县那拉提旅 游风景区	1042.1	喀什地区噶尔老城景区
1	喀什地区噶尔老城景区	218.3	喀什地区泽普县金胡杨景区

2	喀什地区泽普县金胡杨景区	923.1	阿勒泰地区富蕴县可可托海景区
2	阿勒泰地区富蕴县可可托海景区	450.8	阿勒泰地区布尔津县喀纳斯景区
3.5	乌鲁木齐市	2540	西安市

由表 2 中结果可以得出，游爱好者从西安出发，规划设计旅游线路，游遍 201 个 5A 级景区至少需要 12 年。

六、问题二模型的建立、求解及应用

题目要求建立数学模型，设计一个十年游遍所有 201 个 5A 景区、费用最优、旅游体验最好的旅游线路，并给出每一次旅游的具体线路，本题的解题思路如图 4 所示。

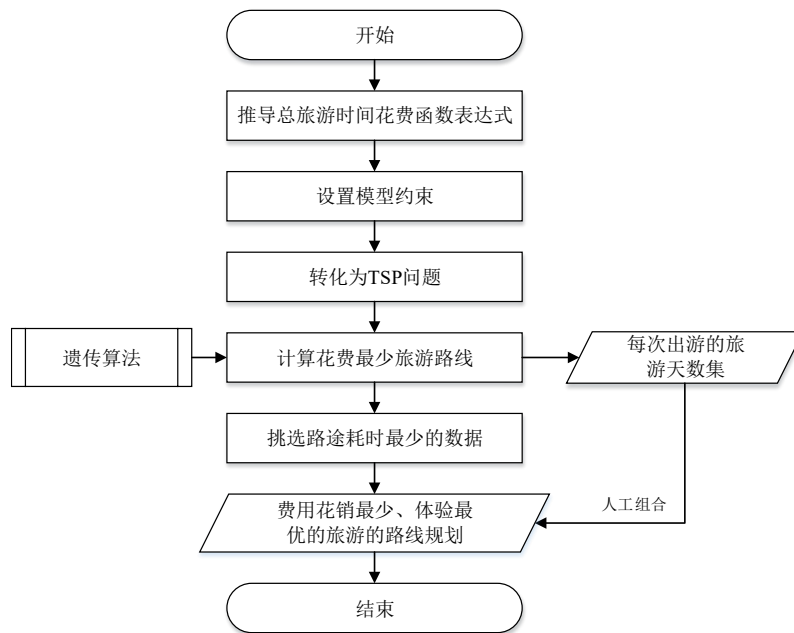


图 4 问题 2 的流程图

6.1 模型的建立

6.1.1 目标函数的确立

经过分析可以知道本题的目标是实现一个十年游遍所有 201 个 5A 景区、费用最优、旅游体验最好的旅游线路，给出每一次旅游的具体线路，我们的做法是先查出这 202 个地点（包括出发地以及 201 个旅游景点），每个地点到其他 201 个地点最经济的交通工具（飞机，高铁，自驾等）及其费用。在满足相应的约束条件下，建立数学模型计算出最佳路线。

游览的总费用包括交通费、各个景点住宿费用总和，由于景点的游览费总和一定，所以，可以仅考虑交通费、各个景点住宿费，从而得到目标函数：

$$\min C = C_T + C_D \quad (11)$$

式中： C_T 为交通费总和； C_D 为住宿费总和。

(1) 总交通费

由题可知，可考虑乘坐高铁或飞机到达与景区相邻的省会城市，而后采用租车的方

式自驾到景区游览。则总交通费应包括总飞机费，总高铁费和总出租车费（自驾），即：

$$C_T = C_p + C_g + C_d \quad (12)$$

式中： C_p 为飞机票费用； C_g 为总高铁费用； C_d 总租车费用（自驾）

①省会到省会：

方式一：省会之间乘坐飞机， P_{mn} 表示从省会 m 到省会 n 的单程飞机票价， $r_{p,mm}^{(k)}$ 表示第 k 次出游是否从省会 m 到省会 n 坐飞机的 0-1 变量。乘坐飞机取 $r_{p,mm}^{(k)} = 1$ ，没有乘坐飞机则 $r_{p,mm}^{(k)} = 0$ ，则飞机票总费用为：

$$C_p = \sum_m \sum_n (P_{mn} \times r_{p,mm}^{(k)})$$

方式二：省会之间乘坐高铁， G_{mn} 表示从省会 m 到省会 n 的单程飞机票价， $r_{G,mm}^{(k)}$ 表示第 k 次出游是否从省会 m 到省会 n 坐高铁的 0-1 变量。乘坐高铁取 $r_{p,mm}^{(k)} = 1$ ，没有乘坐高铁则 $r_{p,mm}^{(k)} = 0$ ，则高铁票总费用为：

$$C_G = \sum_m \sum_n (G_{mn} \times r_{G,mm}^{(k)})$$

②省会到景点：

景点与景点之间出租自驾， d_{aj} 表示为出发地 a 与景点 j 之间的距离，其中高速路的长度为 d_{aj}^g ，普通公路的长度为 d_{aj}^p ，则有：

$$d_{aj} = d_{aj}^g + d_{aj}^p$$

$x_{aj}^{(k)}$ 表示第 k 次出游是否从出发地 a 与景点 j 之间自驾的 0-1 变量。若自驾取 $r_{p,mm}^{(k)} = 1$ ，没有自驾则 $r_{p,mm}^{(k)} = 0$ ，则可以得出：

$$C_d^{g,(k)} = 1 \times \sum_k (x_{aj}^{(k)} \times d_{aj}^g) = \sum_k (x_{aj}^{(k)} \times d_{aj}^g)$$

$$C_d^{p,(k)} = 0.6 \times \sum_k (x_{aj}^{(k)} \times d_{aj}^p) = 0.6 \sum_k (x_{aj}^{(k)} \times d_{aj}^p)$$

高速公路的油耗加过路费平均为 1.00 元/公里，普通公路上油耗平均为 0.60 元/公里，则出租自驾总费用为：

$$C_d^{(k)} = C_d^{g,(k)} + C_d^{p,(k)} = \sum_k (x_{aj}^{(k)} \times d_{aj}^g) + 0.6 \sum_k (x_{aj}^{(k)} \times d_{aj}^p)$$

则总交通费为：

$$C_T = C_p + C_g + C_d = \sum_m \sum_n (P_{mn} \times r_{p,mm}^{(k)}) + \sum_m \sum_n (G_{mn} \times r_{G,mm}^{(k)}) + \sum_k (x_{aj}^{(k)} \times d_{aj}^g) + 0.6 \sum_k (x_{aj}^{(k)} \times d_{aj}^p) \quad (13)$$

(2) 住宿费用

设住在景区和省会的天数为 $D_{s1}^{(k)}$ ，住在地级市天数为 $D_{s2}^{(k)}$ ，住在县城天数为 $D_{s3}^{(k)}$ ，则根据题意住宿费省会城市和旅游景区 200 元/人·天，地级市 150 元/人·天，县城 100 元/人·天，可得出总住宿费（3 人）为：

$$C_D = \sum_k 3 \times (200D_{s1}^{(k)} + 150D_{s2}^{(k)} + 100D_{s3}^{(k)}) = \sum_k (600D_{s1}^{(k)} + 450D_{s2}^{(k)} + 300D_{s3}^{(k)}) \quad (14)$$

从而得到的目标函数为：

$$\begin{aligned}
\min C &= C_T + C_D \\
&= \sum_m \sum_n (p_{mn} \times r_{p,mn}^{(k)}) + \sum_m \sum_n (G_{mn} \times r_{G,mn}^{(k)}) + \sum_k (x_{aj}^{(k)} \times d_{aj}^g) \\
&+ 0.6 \sum_k (x_{aj}^{(k)} \times d_{aj}^p) + \sum_k (600D_{s1}^{(k)} + 450D_{s2}^{(k)} + 300D_{s3}^{(k)})
\end{aligned} \quad (15)$$

6.1.2 约束条件:

将要参观的的景点看成为一个圈，每个景点看做成圈上的一个定点，对于没一个所设的定点只允许最多一条边进入，同样只允许一条边出来，并且只有一条进入，就要有一条边出来，因此可得到一下约束：

$$\sum_{i=0}^{m^{(k)}} x_{ij}^{(k)} = \sum_{j=0}^{m^{(k)}} x_{ij}^{(k)} \leq 1 (l=1,2,3,\dots,m^{(k)})$$

当 $k=0$ 时，

$$\begin{cases} \sum_i x_{ij}^{(0)} = 1 & \text{起点在出发点} \\ \sum_j x_{ij}^{(0)} = 1 & \text{终点在出发点} \end{cases}$$

(1) 时间约束

这里定义 T 为旅游的总时间，显然 $T > 0$

(2) 旅游景点数约束

根据假设，这个旅游路线是环形，即最终游客要回到出发点，因此 $\sum_{i=1}^{202} \sum_{j=1}^{202} r_{ij} = 202$

表示游客旅游的景点数，游客要游玩 202 个景点，所以：

$$\sum_{i=1}^{202} \sum_{j=1}^{202} r_{ij} = 202$$

(3) 0-1 变量约束

由于旅游路线是环形的，把各个景点连成一个圈，把每个景点看成圈上的一个点。对于每个点来说，只允许最多一条边进入，同样只有一条边出来，并且当有一条边进入时必有一条边出来。由此可得约束条件：

$$\sum_i r_{ij} = \sum_j r_{ij} \leq 1 \quad (i, j=1,2,\dots,202)$$

当 $i=1$ 时，游客从出发点出发，所以 $\sum_{i=1} r_{ij} = 1$ ：

当 $j=1$ 时，游客最终回到出发点，所以 $\sum_{j=1} r_{ij} = 1$

综上所述可知，

$$\sum_i r_{ij} = \sum_j r_{ij} \leq 1 \quad (i, j=1,2,\dots,202)$$

$$\sum_{i=1} r_{ij} = 1 \quad \sum_{j=1} r_{ij} = 1$$

同样，当 $i, j \geq 2$ 时，根据题意不可能出现 $r_{ij} = r_{ji} = 1$ ，即不可能出现游客往返旅游，由此可得约束条件：

$$r_{ij} \times r_{ji} = 0$$

(4) 出行时间限制

$t_g^{(k)}$ 表示第 k 次乘坐高铁到达与景区相邻的省会城市所用的时间， $t_{gd}^{(k)}$ 表示到达的时间，要求选择高铁出行要求当天乘坐高铁的时间不超过 6 个小时，乘坐高铁或飞机的当天至多安排半天的景区游览。

$$t_g^{(k)} \leq 6h$$

6.1.3 模型建立：

综上所述，可得到总模型为：

$$\begin{aligned} \min C &= C_T + C_D \\ &= \sum_m \sum_n (p_{mn} \times r_{p,mn}^{(k)}) + \sum_m \sum_n (G_{mn} \times r_{G,mn}^{(k)}) + \sum_k (x_{aj}^{(k)} \times d_{aj}^g) \\ &+ 0.6 \sum_k (x_{aj}^{(k)} \times d_{aj}^p) + \sum_k (600D_{s1}^{(k)} + 450D_{s2}^{(k)} + 300D_{s3}^{(k)}) \end{aligned} \quad (16)$$

约束条件：

$$\left\{ \begin{array}{l} \sum_{i=1}^{202} \sum_{j=1}^{202} r_{ij} = 202 \\ \sum_i r_{ij} = \sum_j r_{ij} \leq 1 (i, j = 1, 2, \dots, 202) \\ \sum_{i=1} r_{ij} = 1, \sum_{j=1} r_{ij} = 1 \\ r_{ij} \times r_{ji} = 0 \\ T > 0 \\ p_i > 0 \\ m_3 = 60 \\ t_g^{(k)} \leq 6h \end{array} \right. \quad (17)$$

6.2 模型求解及分析

问题二为多目标优化模型，在求解时，使用遗传算法求解该问。遗传算法（GA）是基于 Darwin 进化论和 Mendal 遗传学说的一种优化搜索方法，从 20 世纪 60 年代开始兴起。它是基于自然选择和基因遗传学原理的搜索方法，将“优胜劣汰，适者生存”的生物进化原理引入待优化参数形成的编码串群体中，按照一定的适配值函数及一系列遗传操作对各个个体进行筛选，从而使适配值高的个体被保留下来，组成新的群体。新群体包含上一代的大量信息，并且引入了新的优于上一代的个体。这样周而复始，群体中各个体适应度不断提高，直至满足一定条件。此时，群体中适配值最高的个体即为待优

化参数的最优解。

使用遗传算法，可以得到最优的几组费用花销最少的可行解，在这些可行解中，本文挑选出游览时间最少的可行解，即可满足费用花销最少且体验最佳的要求。

6.2.1 适应度函数

本问的适应度函数为

$$f_i = \frac{1}{\sum_{j=1}^n (C_j - C'_j)^2} \quad (18)$$

f_i 表示某个体的适应度， n 为输入输出数据的组数， $|C_j - C'_j|$ 为第 j 组游览的总费与实际的游览的总费之差。

(1) 选择操作

选择是从一个旧种群中选择生命力强的个体位串，产生新种群的过程。本例中选择操作采用适度比例法，它根据某个体的适应度与该代全部个体适应度之和的比值，来决定其子孙遗留的可能性，即在第 n 代中，某个体 i 被选取的概率为：

$$P_{si} = \frac{f_i}{\sum f_N}$$

其中， f_i 为某个体的适应度， $\sum f_N$ 为该代全部个体的适应度之和，可根据式 (18) 来进行计算。确定概率后，可用轮盘赌的选择方法来实现选择操作。例如，用计算机生成 $0 \sim 1$ 之间均匀分布的随机数，若 $P_{si} = 0.5$ ，则当产生的随机数在 $0 \sim 0.5$ 之间时，该串被复制，否则被淘汰。

(2) 交叉操作

交叉运算是指两个相互配对的染色体按某种方式相互交换某部分基因，从而形成两个新的个体。在遗传算法中，将种群中的 N 个个体以随机的方式组成 $N/2$ 对配对个体组，交叉操作在这些配对个体组中的两个个体间进行的。

考察某配对个体组中的两个个体 a 、 b ，交叉操作采用一定方式将它们变为两个新的个体 a' 、 b' 。在遗传算法中，交叉操作过程需要满足：

$$a + b = a' + b'$$

基于上式，浮点数编码的交叉操作采用如下方式来实现：

$$\begin{cases} a' = (1 - \alpha)a + \beta b \\ b' = (1 - \beta)b + \alpha a \end{cases} \quad (19)$$

其中， α 、 β 为区间 $(0, r)$ 上均匀分布的随机数，且有 $r \leq 1$ ，调整 r 的大小即可控制交叉操作的变化范围。

(3) 变异操作

使用变异算子来调整个体编码串中的部分基因值，可以从局部的角度出发使个体更加逼近最优解，提高遗传算法的局部搜索能力。变异操作将某个个体的参数 c ，操作变为域内的另一个值 c' 。浮点数编码的遗传算法采用下式进行：

$$c' = N(c, \sigma) \quad (20)$$

其中， $N(c, \sigma)$ 表示平均值为 c ，方差为 σ 的正态分布随机数。可以看出，变异操作即是以当前值为中心，主要在一个小范围内进行随机扰动的变化。

6.2.2 算法设计

根据以上介绍，就遗传算法本身而言，其步骤主要如下：

- (1) 确定编码方式，选取一定大小的初始种群；
- (2) 计算所选种群中每一个个体的适应度函数值及复制概率；
- (3) 用轮盘赌方法，淘汰相应个数的函数值较小的个体，替换为相应个数的函数值较大的个体；
- (4) 对个体随机两两配对，按指定概率 P_c 进行交叉操作；
- (5) 对每一个体中的每一参数，按指定概率 P_m 进行变异操作；
- (6) 若满足收敛条件则输出最优解并退出，否则返回执行步骤 (2)，如此直到循环终止条件触发。

具体的算法流程如图5示。

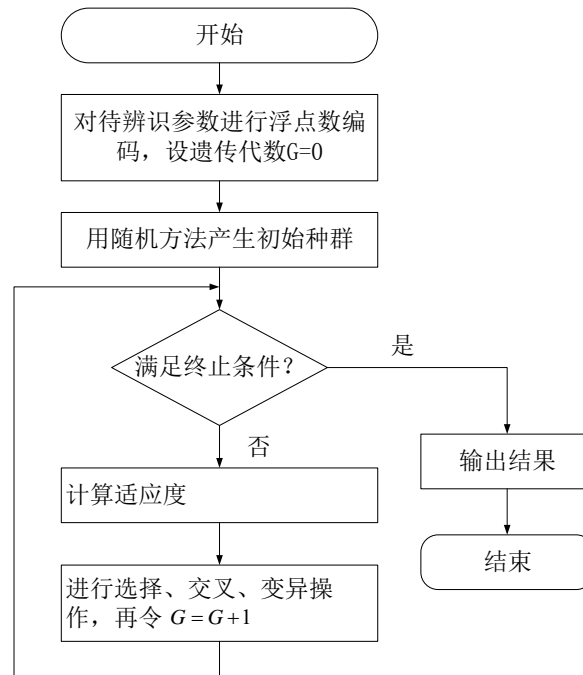


图 5 算法流程图

6.2.3 求解结果及分析

通过算法分析，得出以下结果，如下表 4 和表 5 所示。

表 4 第一条路线西安→太原路线的最低总费用（其他省内旅游路线详见附录）

出发地	出发方式	消耗时间(天)	费用(路 费(元))	路途时 间(h)	游览景区
西安市	租车自驾	0.5	304.6	0.3	西安大雁塔—大唐芙蓉园景区
西安大雁塔—大 唐芙蓉园景区	租车自驾	0.5	322.6	0.3	西安华清池景区
西安华清池景区	租车自驾	0.5	305.5	0.2	西安秦始皇兵马俑博物馆
西安秦始皇兵马 俑博物馆	租车自驾	0.5	382	0.2	宝鸡扶风县法门寺佛文化景区
宝鸡扶风县法门 寺佛文化景区	租车自驾	1	838.1	0.3	渭南华阴市华山风景区
渭南华阴市华山 风景区	租车自驾	1	882	4	延安黄陵县黄帝陵景区

西安市	高铁	0.9	535.5	3	太原市
总和 1		4.9	3570.3	8.3	

表 5 全国路线和费用

路线	是否回西安	省内景点交通方式	省际交通方式	费用	消耗时间(天)
西安→太原	是	租车自驾	高铁+高铁	3570.3	4.9
太原→呼和浩特	否	租车自驾	飞机	3431.4	4.1
呼和浩特→天津	否	租车自驾	飞机	3101.6	2.1
天津→西安→郑州	是	租车自驾	飞机+高铁	4504.2	2
郑州→西安→石家庄	是	租车自驾	高铁+高铁	8579.5	11.4
石家庄→银川	否	租车自驾	飞机	6103	5.6
银川→西安→长沙	是	租车自驾	飞机+高铁	5504.2	3.1
长沙→上海	否	租车自驾	飞机	8351.1	6.1
上海→西安→北京	是	租车自驾	飞机+高铁	6925.2	3.4
北京→长春	否	租车自驾	飞机	5348.4	4.1
长春→西安→武汉	是	租车自驾	飞机+高铁	7875.9	3.5
武汉→西安→西宁	是	租车自驾	高铁+飞机	11109.6	10.8
西宁→海口	否	租车自驾	飞机	2781.8	1.5
海口→西安→成都市	是	租车自驾	飞机+飞机	9363.6	2.9
成都→西安→济南	是	租车自驾	飞机+飞机	12742.8	10.2
济南→西安→贵阳	是	租车自驾	飞机+飞机	10305.9	6.2
贵阳→南宁	否	租车自驾	飞机	5296.1	3.6
南宁→西安→兰州	是	租车自驾	飞机+飞机	10850.8	3.8
兰州→西安→南昌	是	租车自驾	飞机+飞机	10249	5.3
南昌→西安→合肥	是	租车自驾	飞机+飞机	11159.6	8.3
合肥→西安→重庆	是	租车自驾	飞机+飞机	10816	8.2
重庆→西安→杭州	是	租车自驾	飞机+飞机	14326.8	8.8
杭州→西安→沈阳	是	租车自驾	飞机+飞机	14118.9	9.4
沈阳→西安→福州	是	租车自驾	飞机+飞机	10946.8	3.4
福州→西安→昆明	是	租车自驾	飞机+飞机	15453	8.9
昆明→西安→广州	是	租车自驾	飞机+飞机	13578.4	6.4
广州→西安→哈尔滨	是	租车自驾	飞机+飞机	14794	8
哈尔滨→西安→拉萨	是	租车自驾	飞机+飞机	17858.5	6.7
拉萨→西安→乌鲁木齐	是	租车自驾	飞机+飞机	13173	1.8
乌鲁木齐→西安→南京	是	租车自驾	飞机+飞机	14778.8	12.2
南京→西安	是	租车自驾	飞机	10504.9	12
总和				297503.1	8 年(换算)

上表 5 所示，本文建立的数学模型游遍所有 201 个 5A 景区所需的时间为 8 年，符合题目要求的 10 年内，并且满足费用最低，旅游体验最好的旅游线路，本文得出所用

的总费用为 29.75 万元。

七 问题三模型的建立、求解及应用

经过分析可以知道本题的目标是在第二问的基础上加以推广,实现一个十年游遍所有 201 个 5A 景区、费用最优、旅游体验最好的旅游线路,给出每一次旅游的具体线路。与第二问不同的是,我们需要建立一个更加普适的模型,这个模型可以全国自驾游的爱好者都通用。那么,需要将模型一些特定的条件加以修改,是之更具通用性。我们做出如下的约定:

1. 针对于不同的旅行时间和不同的旅行城市,还应该考虑到一日三餐的餐饮费(餐饮费简化为省会城市和旅游景区 100 元/人·天,地级市 80 元/人·天,县城 50 元/人·天,餐饮费中包括了当天的饮料、饮用水、水果等食物的费用)。

2. 同时,还应该考虑每个景点的门票费(C_i)。

在问题二的基础上,加入上述的约定之后,可以建立更加通用的模型。

其思路图如下图 6 示:

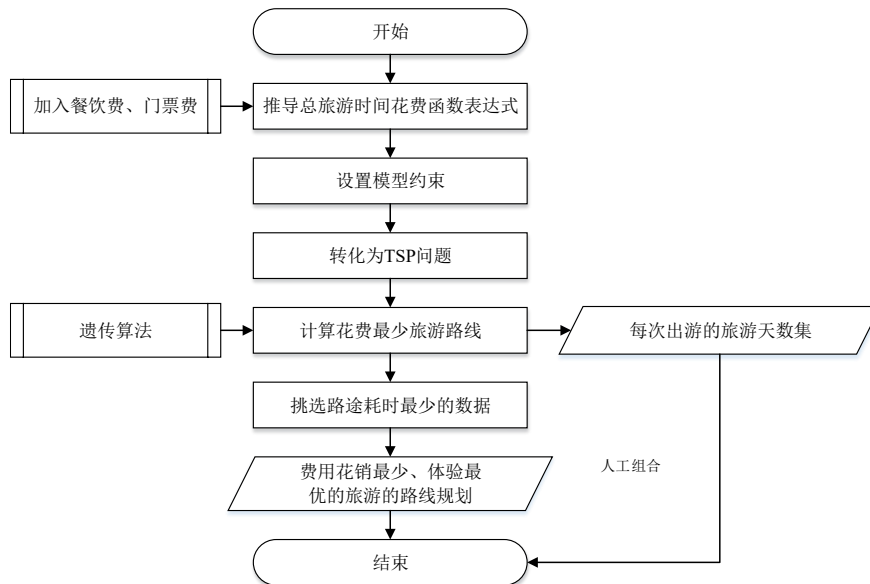


图 6 问题 3 的流程图

7.1 模型的建立

7.1.1 目标函数的确立

与问题二不同,游览的总费用包括交通费、各个景点住宿费、景点门票费以及餐饮费的总和,从而得到目标函数:

$$\min C = C_T + C_D + C_R + C_i \quad (21)$$

式(21)中: C_T 为交通费总和; C_D 为住宿费总和, C_R 为餐饮费总和, C_i 为景点 i 的门票费。

(1) 总交通费

和问题二一致,总交通费主要由飞机票费用、总高铁费用和总的租车费用组成。可考虑乘坐高铁或飞机到达与景区相邻的省会城市,而后采用租车的方式自驾到景区游览。则总交通费应包括总飞机费,总高铁费和总出租车费(自驾),即

$$C_T = C_p + C_g + C_d \quad (22)$$

式中： C_p 为飞机票费用； C_g 为总高铁费用； C_d 总租车费用（自驾）。

方式一：

省会之间乘坐飞机， P_{mn} 表示从省会 m 到省会 n 的单程飞机票价， $r_{p,mm}^{(k)}$ 表示第 k 次出游是否从省会 m 到省会 n 坐飞机的 0-1 变量。乘坐飞机取 $r_{p,mm}^{(k)} = 1$ ，没有乘坐飞机则 $r_{p,mm}^{(k)} = 0$ ，则飞机票总费用为：

$$C_p = \sum_m \sum_n (P_{mn} \times r_{p,mm}^{(k)})$$

方式二：省会之间乘坐高铁， G_{mn} 表示从省会 m 到省会 n 的单程飞机票价， $r_{G,mm}^{(k)}$ 表示第 k 次出游是否从省会 m 到省会 n 坐高铁的 0-1 变量。乘坐高铁取 $r_{p,mm}^{(k)} = 1$ ，没有乘坐高铁则 $r_{p,mm}^{(k)} = 0$ ，则高铁票总费用为：

$$C_G = \sum_m \sum_n (G_{mn} \times r_{G,mm}^{(k)})$$

方式三：景点与景点之间出租自驾， d_{aj} 表示为出发地 a 与景点 j 之间的距离，其中高速路的长度为 d_{aj}^g ，普通公路的长度为 d_{aj}^p ，则有：

$$d_{aj} = d_{aj}^g + d_{aj}^p$$

$x_{aj}^{(k)}$ 表示第 k 次出游是否从出发地 a 与景点 j 之间自驾的 0-1 变量。若自驾取 $r_{p,mm}^{(k)} = 1$ ，没有自驾则 $r_{p,mm}^{(k)} = 0$ ，则可以得出：

$$C_d^{g,(k)} = 1 \times \sum_k (x_{aj}^{(k)} \times d_{aj}^g) = \sum_k (x_{aj}^{(k)} \times d_{aj}^g)$$

$$C_d^{p,(k)} = 0.6 \times \sum_k (x_{aj}^{(k)} \times d_{aj}^p) = 0.6 \sum_k (x_{aj}^{(k)} \times d_{aj}^p)$$

高速公路的油耗加过路费平均为 1.00 元/公里，普通公路上油耗平均为 0.60 元/公里，则出租自驾总费用为：

$$C_d^{(k)} = C_d^{g,(k)} + C_d^{p,(k)} = \sum_k (x_{aj}^{(k)} \times d_{aj}^g) + 0.6 \sum_k (x_{aj}^{(k)} \times d_{aj}^p)$$

则总交通费为：

$$\begin{aligned} C_T &= C_p + C_g + C_d \\ &= \sum_m \sum_n (P_{mn} \times r_{p,mm}^{(k)}) + \sum_m \sum_n (G_{mn} \times r_{G,mm}^{(k)}) + \sum_k (x_{aj}^{(k)} \times d_{aj}^g) + 0.6 \sum_k (x_{aj}^{(k)} \times d_{aj}^p) \end{aligned} \quad (23)$$

(2) 住宿费用

设住在景区和省会的天数为 $D_{s1}^{(k)}$ ，住在地级市天数为 $D_{s2}^{(k)}$ ，住在县城天数为 $D_{s3}^{(k)}$ ，则根据题意可得出总住宿费为（一个旅游者）：

$$C_D = \sum_k (200D_{s1}^{(k)} + 150D_{s2}^{(k)} + 100D_{s3}^{(k)}) \quad (24)$$

(3) 餐饮费

旅行者有可能早上、晚上并不在同一个城市，为简化计算，以晚上城市或者景区的标准为当天的饮食标准。设第 k 次出游在景区和省会饮食的天数为 $R_{s1}^{(k)}$ ，在地级市饮食

的天数为 $R_{s_2}^{(k)}$ ，在县城饮食的天数为 $R_{s_3}^{(k)}$ 。则根据题意可得出总住宿费为：

$$C_R = \sum_k (100R_{s_1}^{(k)} + 80R_{s_2}^{(k)} + 50R_{s_3}^{(k)}) \quad (25)$$

(4) 门票费

总共有 201 个景点，所以 $i = 1, 2, \dots, 201$ 。那么，景点的总费用为

$$C_I = \sum_i C_i$$

从而得到的目标函数为：

$$\begin{aligned} \min C &= C_T + C_D + C_R \\ &= \sum_m \sum_n (P_{mn} \times r_{p, mn}^{(k)}) + \sum_m \sum_n (G_{mn} \times r_{G, mn}^{(k)}) + \sum_k (x_{aj}^{(k)} \times d_{aj}^g) \\ &\quad + 0.6 \sum_k (x_{aj}^{(k)} \times d_{aj}^p) + \sum_k (200D_{s_1}^{(k)} + 150D_{s_2}^{(k)} + 100D_{s_3}^{(k)}) \\ &\quad + \sum_k (100R_{s_1}^{(k)} + 80R_{s_2}^{(k)} + 50R_{s_3}^{(k)}) + \sum_i C_i \end{aligned} \quad (26)$$

7.1.2 约束条件：

与问题二类似，将要参观的的景点看成为一个圈，每个景点看做成圈上的一个定点，对于没一个所设的定点只允许最多一条边进入，同样只允许一条边出来，并且只有一条进入，就要有一条边出来，因此可得到一下约束：

$$\sum_{i=0}^{m^{(k)}} x_{ij}^{(k)} = \sum_{j=0}^{m^{(k)}} x_{ij}^{(k)} \leq 1 (l = 1, 2, 3, \dots, m^{(k)})$$

当 $k = 0$ 时， $\begin{cases} \sum_i x_{ij}^{(0)} = 1 & \text{起点在出发点} \\ \sum_j x_{ij}^{(0)} = 1 & \text{终点在出发点} \end{cases}$

(1) 时间约束

这里定义 T 为旅游的总时间，显然 $T > 0$

(2) 旅游景点数约束

根据假设，这个旅游路线是环形，即最终游客要回到出发点，因此 $\sum_{i=1}^{202} \sum_{j=1}^{202} r_{ij} = 202$

表示游客旅游的景点数，游客要游玩 202 个景点，所以：

$$\sum_{i=1}^{202} \sum_{j=1}^{202} r_{ij} = 202$$

(3) 0-1 变量约束

由于旅游路线是环形的，把各个景点连成一个圈，把每个景点看成圈上的一个点。对于每个点来说，只允许最多一条边进入，同样只有一条边出来，并且当有一条边进入时必有一条边出来。由此可得约束条件：

$$\sum_i r_{ij} = \sum_j r_{ij} \leq 1 \quad (i, j = 1, 2, \dots, 202)$$

当 $i=1$ 时, 游客从出发点出发, 所以 $\sum_{i=1} r_{ij} = 1$:

当 $j=1$ 时, 游客最终回到出发点, 所以 $\sum_{j=1} r_{ij} = 1$

综上所述,

$$\sum_i r_{ij} = \sum_j r_{ij} \leq 1 \quad (i, j = 1, 2, \dots, 202)$$

$$\sum_{i=1} r_{ij} = 1 \quad \sum_{j=1} r_{ij} = 1$$

同样, 当 $i, j \geq 2$ 时, 根据题意不可能出现 $r_{ij} = r_{ji} = 1$, 即不可能出现游客往返旅游, 由此可得约束条件:

$$r_{ij} \times r_{ji} = 0$$

(4) 出行时间限制

$t_g^{(k)}$ 表示第 k 次乘坐高铁到达与景区相邻的省会城市所用的时间, $t_{gd}^{(k)}$ 表示到达的时间, 要求选择高铁出行要求当天乘坐高铁的时间不超过 6 个小时, 乘坐高铁或飞机的当天至多安排半天的景区游览。

$$t_g^{(k)} \leq 6h$$

(5) 门票价格约束

$$C_I = \sum_i C_i \geq 0$$

7.1.3 模型建立:

综上所述, 可得到总模型为:

$$\begin{aligned} \min C &= C_T + C_D + C_R \\ &= \sum_m \sum_n (P_{mn} \times r_{p,mn}^{(k)}) + \sum_m \sum_n (G_{mn} \times r_{G,mn}^{(k)}) + \sum_k (x_{aj}^{(k)} \times d_{aj}^g) \\ &\quad + 0.6 \sum_k (x_{aj}^{(k)} \times d_{aj}^p) + \sum_k (200D_{s1}^{(k)} + 150D_{s2}^{(k)} + 100D_{s3}^{(k)}) \\ &\quad + \sum_k (100R_{s1}^{(k)} + 80R_{s2}^{(k)} + 50R_{s3}^{(k)}) + \sum_i C_i \end{aligned} \quad (27)$$

约束条件:

$$\begin{cases}
\sum_{i=1}^{202} \sum_{j=1}^{202} r_{ij} = 202 \\
\sum_i r_{ij} = \sum_j r_{ij} \leq 1 (i, j = 1, 2, \dots, 202) \\
\sum_{i=1} r_{ij} = 1, \sum_{j=1} r_{ij} = 1 \\
r_{ij} \times r_{ji} = 0 \\
T > 0 \\
p_i > 0 \\
t_g^{(k)} \leq 6h \\
C_I = \sum_i C_i \geq 0
\end{cases} \quad (28)$$

7.2 模型求解及分析

和问题二类似，本题同样可以采用遗传算法来求解，在此不再赘述。

7.2.3 求解结果及分析

针对问题 2 所建立的模型基础上加以推广，本文所建的模型可以为全国的自驾游爱好者规划设计类似的旅游线路，下面的是以常住地在北京市的自驾游爱好者的十年旅游计划结果，如下表 6 和表 7 所示：

表 6 出发地为北京的路线（其他省内旅游路线详见附录）

消耗时间 (天)	出发方式	出发地	费用 (元)	路途时间 (h)	游览景区
0.5	租车自驾	北京市	301	0.3	故宫博物院
0.5	租车自驾	故宫博物院	302.9	0.3	恭王府景区
0.5	租车自驾	恭王府景区	305.1	0.2	天坛公园
0.5	租车自驾	天坛公园	311.9	0.2	北京奥林匹克公园(鸟巢-水立方-中国科技馆-国家奥林匹克森林公园)
0.5	租车自驾	北京奥林匹克公园(鸟巢-水立方-中国科技馆-国家奥林匹克森林公园)	307.7	0.3	颐和园
0.5	租车自驾	颐和园	321.5	0.2	明十三陵景区(神路-定陵-长陵-昭陵)
0.5	租车自驾	明十三陵景区(神路-定陵-长陵-昭陵)	348.3	0.1	八达岭—慕田峪长城旅游区
0.6	高铁	北京市	163.5	0.7	天津市

表 7 全国路线和费用

路线	是否回北	省内景点交通方式	省际交通方式	费用	消耗时间 (天)
----	------	----------	--------	----	-------------

	京				
北京→天津	是	租车自驾	高铁	2361.9	4.1
天津→石家庄	否	租车自驾	高铁	577.2	2
石家庄→北京→济南	是	租车自驾	高铁+高铁	4162	6.9
济南→上海	否	租车自驾	飞机+高铁	5621.4	7.8
上海→北京→郑州	是	租车自驾	高铁+高铁	3812.7	3.6
郑州→北京→合肥	是	租车自驾	高铁	8894.5	12.1
合肥→北京→西安	是	租车自驾	高铁+高铁	6865	10.3
西安→太原	否	租车自驾	高铁	3570.3	4.9
太原→北京→长沙	是	租车自驾	飞机+高铁	6792.9	4.7
长沙→呼和浩特	否	租车自驾	飞机	7721.1	6.7
呼和浩特→北京→杭州	是	租车自驾	飞机+高铁	4499.6	3.4
杭州→北京→长春	是	租车自驾	高铁+飞机	10806.9	10.4
长春→哈尔滨	否	租车自驾	高铁	1445.4	2.6
哈尔滨→北京→银川	是	租车自驾	飞机+飞机	13178.5	6.9
银川→西宁	否	租车自驾	飞机	3090.7	2.6
西宁→北京→兰州	是	租车自驾	飞机+飞机	10305.9	6.2
贵阳→南宁	否	租车自驾	飞机	5296.1	3.6
南宁→西安→兰州	是	租车自驾	飞机+飞机	10171.8	2.9
兰州→北京→南昌	是	租车自驾	飞机+飞机	12679	5.9
南昌→北京→沈阳	是	租车自驾	飞机+飞机	13649.6	9.8
沈阳→北京→重庆	是	租车自驾	飞机+飞机	11186.8	3.8
重庆→北京→福州	是	租车自驾	飞机+飞机	16846.8	9.5
福州→北京→广州	是	租车自驾	飞机+飞机	14118.9	9.4
广州→北京→贵阳	是	租车自驾	飞机+飞机	16324	9.1
贵阳→北京→南宁	是	租车自驾	飞机+飞机	15196.1	4.6
南宁→北京→海口	是	租车自驾	飞机+飞机	15950.8	5.2
海口→北京→拉萨	是	租车自驾	飞机+飞机	17493.6	3.9
拉萨→北京→南京	是	租车自驾	飞机+高铁	10723.5	2.5
南京→北京→武汉	是	租车自驾	飞机+飞机	11314.9	12.2
武汉→北京→成都	是	租车自驾	飞机+飞机	16638.6	10.3
成都→北京→昆明	是	租车自驾	飞机+飞机	14182.8	10.4
昆明→北京→乌鲁木齐	是	租车自驾	飞机+飞机	12228.4	6.4
乌鲁木齐→北京	是	租车自驾	飞机	11238.8	12
总和				318946.5	9年（换算）

上表 7 所示，本文建立的数学模型游遍所有 201 个 5A 景区所需的时间为 9 年，符合题目要求的 10 年内，并且满足费用最低，旅游体验最好的旅游线路，最终本文得出所用的总费用为 31.895 万元。

7.3 模型结果的建议

7.3.1 给旅游相关部门的建议

(1) 完善景区之间的高速公路建设。从前三问的计算结果可以看出，很多自然景区由于地处偏远，仅仅只是有专线公路。专线公路连接的往往是景区及其所属城市，如果自驾游爱好者或者其他的旅游者试图去其它的景区旅游，则必须回到景区所属城市，这给旅游者增加了很大一部分的交通耗时和交通费用。因此，完善景点与景点之间的高速公路建设十分有必要。

(2) 扩大景点的停车场面积，尤其在热门景点且人群拥挤的地方，拥有大面积的停车场，将大大提高自驾车旅游的乘客的心里感受。

7.3.2 给旅游者的相关建议

(1) 尽量延长每次出游时间

旅游者每次出行，必然由其常住地（假设为北京）出发，旅游一圈然后返回北京。如果将每年旅游的时间分割成若干时间段，势必造成往返常住地和景点之间的交通费和时间花费增加。因此，将每次出游的时间延长，接近 15 天，这样能够减少出游的次数，同时还能节省交通费和往返时间花费。

(2) 可以考虑组团自驾游

在自驾游的过程中，油费往往占据了交通费的很大一部分。自驾游爱好者可以考虑组团自驾游，这样可以节约一定的油费、过路费等。

(3) 避开旅游热季

由于在设计旅游路线的时候，并没有根据具体景点和景区的旅游热季等问题，所以旅游者在确定具体出行时间时应该选择在景点淡季出游这样能够避免拥挤及因为景区人数过多导致的物价提高带来的总支出费用的增多。

八 问题四模型的建立、求解及应用

自 2007 年 3 月 7 日至 2015 年 7 月 13 日，全国旅游景区质量等级评定委员会分 29 批共批准了 201 家景区为国家 5A 级旅游景区。合理地规划该旅游爱好者的十年旅游计划。从附件 6 可以看出：

2015 年 4 月 2 日上午，国家旅游局通报，从 2014 去年四季度到 2015 年一季度，山西忻州五台山、南京夫子庙秦淮河观光带等 9 家 5A 景区被警告，如下表 8 所示，限期整改，否则可能面临摘牌或降级。

表 8 被警告的 8 家 5A 级旅游景区

河北省保定安新白洋淀景区	山西忻州市五台山风景名胜	黑龙江哈尔滨市太阳岛公园
江苏南京夫子庙秦淮风光带景区	浙江杭州市千岛湖风景名胜	山东枣庄市台儿庄古城景区
河南洛阳市龙潭大峡谷景区	陕西渭南市华山景区	广东清远连州地下河旅游景区

因为要为旅游者设计一个长久的旅行计划，那么必须确定可能的 5A 景区增补。在问题四所给出的附件 6 中给出了 5A 景区评定的方法及指标，附件 7 中则给出了各个省市的 4A 景区。我们假设附件 6 中所给的被警告的 9 个 5A 景区在未来可能被摘牌，那么我们需要从附件 7 中筛选出可能作为被升级为 5A 景区的 4A 景区。

根据附件 6 中所给的景区评价指标，我们考虑建立层次分析法模型来进行 4A 景区升级的筛选问题。其思路图如图 7 所示。

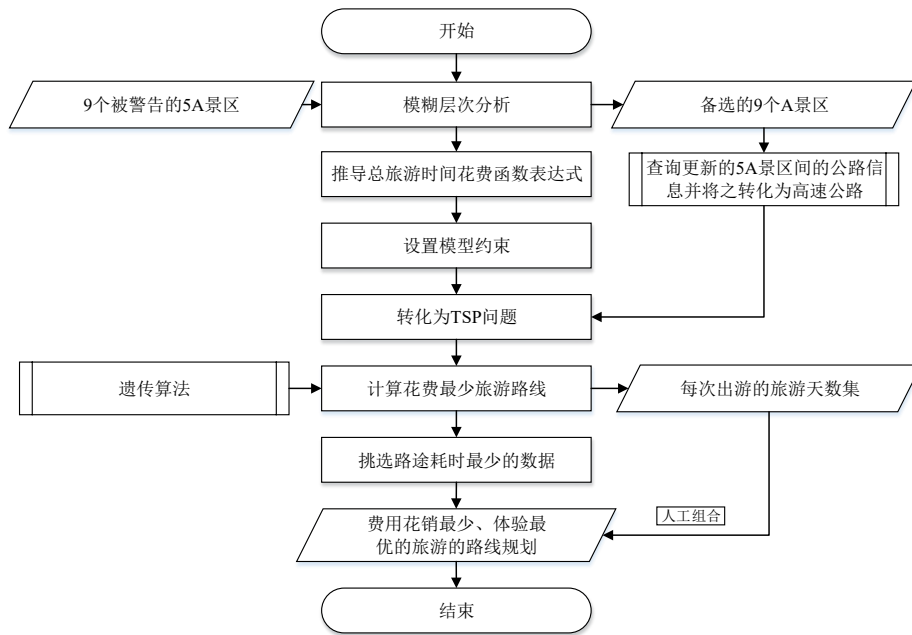


图 7 问题 4 的流程图

8.1 模型假设

1. 假设被警告的 9 个 5A 景区在未来被摘牌；
2. 升级的 4A 景区从被摘牌的 9 个 5A 景区所在省中选取。

8.2 模型建立

5A 景区的评价细则主要分为 3 部分，即服务质量与环境质量、景观质量、游客意见三部分。服务质量与环境质量评分细则（共计 1000 分；5A 不低于 950 分），景观质量评分细则（共计 100 分；5A 不低于 90 分），游客意见评分细则（共计 100 分；5A 不低于 90 分）。

其中，服务质量与环境质量评可以分为旅游交通（130 分）、游览（235 分）、旅游安全（80 分）、卫生（140 分）、邮电服务（20 分）、旅游购物（50 分）、综合管理（200 分）、资源与环境保护（145 分）；景观质量分为资源吸引力（65 分）、市场影响力（35 分）；游客意见分为外部交通、内部游览线路、观景设施、路标指示、景物介绍牌、宣传资料、导游讲解、服务质量、安全保障、环境卫生、厕所、邮电服务、商品购物、餐饮或食品、旅游秩序、景物保护等 16 项，每小项各 5 分，共计 80 分，总体印象（20 分）。本文采用改进模糊层次分析法对此进行研究^[8]，综合评价指标体系的建立如图 8 示

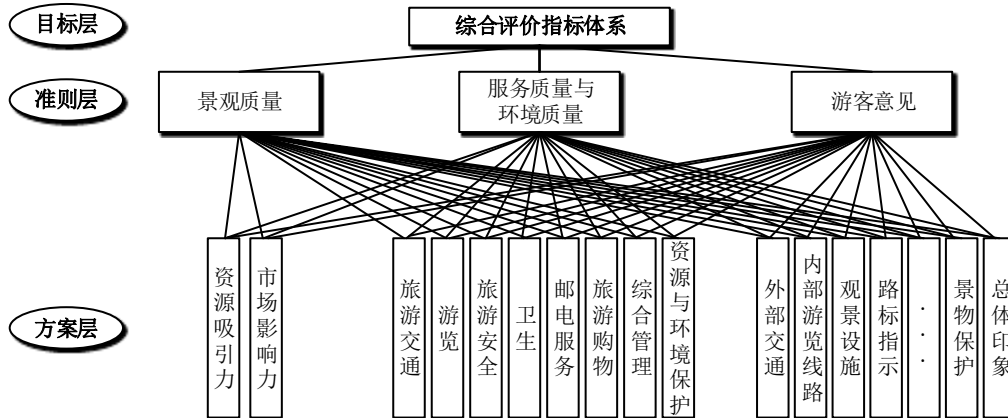


图 8 改进模糊层次分析法综合评价指标体系

在模糊层次分析中，做因素间的两两比较判断，采用一个因素比另一个因素的重要程度来定量表示，则得到模糊判断矩阵 $A = (a_{ij})_{n \times n}$ ，如满足：

- (1) $a_{ii} = 0.5, i = 1, 2, \dots, n$
- (2) $a_{ij} + a_{ji} = 1, i = 1, 2, \dots, n$

则称为模糊互补判断矩阵。通常采用表 9 所示的 0.1~0.9 标度法给予数量标度。

表 9 0.1~0.9 标度法及其意义

标度	定义	说明
0.5	同等重要	两因素相比较，同等重要
0.6	稍微重要	两因素相比较，一个因素比另一个稍微重要
0.7	明显重要	两因素相比较，一个因素比另一个明显重要
0.8	重要得多	两因素相比较，一个因素比另一个重要得多
0.9	极端重要	两因素相比较，一个因素比另一个极端重要
0.1, 0.2, 0.3, 0.4	反比较	若因素 $a_i/a_j = r_{ij}$ ，则与 a_i/a_j 相比较 $a_j/a_i = 1 - r_{ij}$

8.2.1 数学模型的建立

对因素 a_1, a_2, \dots, a_n 相互进行比较得到如下模糊互补判断矩阵：

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \quad (29)$$

步骤 1：设模糊判断矩阵对矩阵按行求和 $r_i = \sum_{k=1}^n a_{ik}, (i = 1, 2, \dots, n)$ ，并实施如下数学代换：

$$r_{ij} = \frac{r_i - r_j}{2(n-1)} + 0.5 \quad (30)$$

得到模糊一致性矩阵 $R = (r_{ij})_{n \times n}$ 。

步骤 2：由矩阵采用行和归一化求得排序向量

$$W = (w_1, w_2, \dots, w_n)^T \quad (31)$$

满足

$$W_i = \frac{\sum_{j=1}^n a_{ij}}{\sum_{i=1}^n \sum_{j=1}^n a_{ij}} (i, j = 1, 2, \dots, n) \quad (32)$$

式 (32) 为求解模糊互补判断矩阵权重向量的公式。

令全体阶模糊互补判断矩阵构成的集合为 G_n 。设 $A = (a_{ij})_{n \times n}$ 和 $B = (b_{ij})_{n \times n} \in G_n$ ，用

范数 $A - B = \sum_{i=1}^n \sum_{j=1}^n |a_{ij} - b_{ij}|$ 表示之间的距离，记为 $\rho(A, B) = A - B$ 。

步骤 3: 模糊互补判断矩阵的一致性检验

由式 (32) 得到的权重值是否合理，还应该进行比较判断的一致性检验。

定义：矩阵 $A = (a_{ij})_{n \times n}$ 和 $B = (b_{ij})_{n \times n}$ 均为模糊判断矩阵，称

$$I(A, B) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n |a_{ij} - b_{ij}| \quad (33)$$

式 (33) 为和的相容性指标。

$W = (w_1, w_2, \dots, w_n)^T$ 是模糊判断矩阵的权重向量，其中 $\sum_{i=1}^n W_i = 1$ ，

$w_i \geq 0 (i = 1, 2, \dots, n)$ ，令 $w_{ij} = \frac{w_i}{w_i + w_j} (\forall i, j = 1, 2, \dots, n)$ ，则称 n 阶矩阵

$$W^* = (W_{ij})_{n \times n} \quad (34)$$

式 (34) 为判断矩阵 A 的特征矩阵。

对于决策者的态度 α ，当相容性指标 $I(A, W^*) \leq \alpha$ 时，认为判断矩阵为满意一致性的。 α 越小则表明决策者对模糊判断矩阵的一致性要求越高，本文取 $\alpha = 0.1$ 。

对于实际问题，一般都是由多个（设 k 个， $k = 1, 2, \dots, m$ ）专家给出同一因素集 x 上的两两比较判断矩阵

$$A_k = (a_{ij}^{(k)})_{m \times m}, k = 1, 2, \dots, m \quad (35)$$

它们均是模糊互补判断矩阵，则可分别得到权重集的集合：

$$W_k = (W_1^{(k)}, W_2^{(k)}, \dots, W_n^{(k)}), k = 1, 2, \dots, m \quad (36)$$

则进行模糊互补判断矩阵的一致性检验，要做以下两个方面的工作：

(1) 检验 m 个矩阵 A_k 的满意一致性

$$I(A_k, W^{(k)}) \leq \alpha, k = 1, 2, \dots, m \quad (37)$$

(2) 检验判断矩阵间的满意相容性

$$I(A_k, A_l) \leq \alpha, k \neq l, k, l = 1, 2, \dots, m \quad (38)$$

可以证明，在模糊互补判断矩阵 $A_k (k = 1, 2, \dots, m)$ 是一致可接受的情况下，它们的综合判断矩阵也是一致可接受的。即只要当式(2-10)和式(2-11)满足时， m 个观众集的均值作为因素集 X 的权重分配向量时合理和可靠的，权重向量表达式

$$W = (W_1, W_2, \dots, W_n) \quad (39)$$

满足：

$$W_i = \frac{1}{n} \sum_{k=1}^m W_i^{(k)}, i=1,2,\dots,n. \quad (40)$$

8.3 模型求解

8.3.1 权重矩阵的确定

从 5A 级景区的评定细则我们可以为模糊矩阵进行打分。服务质量与环境质量评分细则（共计 1000 分；5A 不低于 950 分），景观质量评分细则（共计 100 分；5A 不低于 90 分），游客意见评分细则（共计 100 分；5A 不低于 90 分）。打分的方法分别对各因素作两两比较判断，如表 10 所示，得到权重模糊互补判断矩阵 A 为：

表 10 模糊互补判断矩阵

A	景观质量	服务质量与 环境质量	游客意见
景观质量	0.5	0.1	0.5
服务质量与环境质量	0.9	0.5	0.9
游客意见	0.5	0.1	0.5

根据上式，可得其权重向量为：

$$W = (0.2444 \quad 0.2722 \quad 0.2167)$$

根据上式，A 的特征矩阵计算结果为：

$$A = \begin{bmatrix} 0.5 & 0.2652 & 0.3698 \\ 0.2652 & 0.5 & 0.6254 \\ 0.3698 & 0.6254 & 0.5 \end{bmatrix}$$

根据式 (35) 和式 (36)，A 与 W^* 的相容性指标为： $I(A, W^*) = 0.0584 < 0.1$ ，故可以认为模糊判断矩阵 A 是满意一致的，故其权重 W 分配合理。

8.3.2 建立评判矩阵

本文以海南省为例，评判其省内的 8 个 4A 级景区。如表 11 所示为海南省所有的 8 个 4A 级景区。

表 11 海南省的 8 个 A 级景区

序号	景区名称
1	三亚蜈支洲岛度假中心
2	海口石山火山群国家地质公园
3	海南热带野生动植物园
4	天涯海角风景区
5	三亚亚龙湾国家旅游度假区
6	兴隆热带植物园
7	海南热带海洋世界
8	陵水南湾猴岛生态景区

建立的评判矩阵为：

$$R = \begin{bmatrix} 0.6787 & 0.3922 & 0.7060 & 0.0462 & 0.6948 & 0.0344 & 0.7655 & 0.4898 \\ 0.7577 & 0.6555 & 0.0318 & 0.0971 & 0.3171 & 0.4387 & 0.7952 & 0.4456 \\ 0.7431 & 0.1712 & 0.2769 & 0.8235 & 0.9502 & 0.3816 & 0.1869 & 0.6463 \end{bmatrix}$$

8.3.3 方案总排序计算结果：

$$B = W \times R = \begin{bmatrix} 0.6787 & 0.3922 & 0.7060 & 0.0462 & 0.6948 & 0.0344 & 0.7655 & 0.4898 \\ 0.7577 & 0.6555 & 0.0318 & 0.0971 & 0.3171 & 0.4387 & 0.7952 & 0.4456 \\ 0.7431 & 0.1712 & 0.2769 & 0.8235 & 0.9502 & 0.3816 & 0.1869 & 0.6463 \end{bmatrix}$$

$$= (0.5332 \quad 0.3114 \quad 0.2412 \quad 0.2162 \quad 0.4620 \quad 0.2105 \quad 0.4440 \quad 0.3810)$$

按最大隶属度原则，备选 5A 景区优选序为：1 > 5 > 7 > 8 > 2 > 3 > 4 > 6，最终确定可以升级为 5A 景区的备选 4A 级景区为三亚蜈支洲岛度假中心。

按照海南省的计算方法，可以给出 9 个省区的备选 5A 级景区，如表 12 所示。

表 12 原 5A 景区和备选 5A 景区列表

序号	可能摘牌的 5A 景区	备选 5A 景区	所属省份
1	河北省保定安新白洋淀景区	→ 石家庄市灵寿秋山景区	河北
2	山西忻州市五台山风景名胜	→ 临汾市尧庙-华门旅游区	山西
3	黑龙江哈尔滨市太阳岛公园	→ 鹤岗市萝北名山景区	黑龙江
4	江苏南京夫子庙秦淮风光带	→ 虞山-尚湖风景区	江苏
5	浙江杭州市千岛湖风景名胜	→ 绍兴柯岩风景区	浙江
6	山东枣庄市台儿庄古城景区	→ 青岛天泰温泉度假区	山东
7	河南洛阳市龙潭大峡谷景区	→ 南阳武侯祠	河南
8	陕西渭南市华山景区	→ 榆林佳县白云山景区	陕西
9	广东清远市连州地下河旅游	→ 梅州灵光寺风景区	广东

8.4 模型和算法

题 4 的模型和算法与上文的一样，本文取景区较多的城市南京为出发点，将数据代入模型中，得到较为合理的结果，即规划该旅游爱好者的十年旅游计划。

8.5 算法结果

通过计算，假设该旅游爱好者从南京出发，则他的十年旅游计划如下表 13 所示：

表 13 优化后的十年旅游计划

路线	是否回南京	省内景点交通方式	省际交通方式	费用	消耗时间(天)
南京→合肥	否	租车自驾	高铁	7146.4	12.6
合肥→上海	否	租车自驾	高铁	577.2	9.3
上海→南京→杭州	是	租车自驾	高铁+高铁	1964.7	2.9
杭州→南京→长沙	是	租车自驾	高铁+高铁	6629.4	10.6
长沙→天津	否	租车自驾	飞机	7511.1	6.7
天津→南京→郑州	是	租车自驾	飞机+高铁	3772.2	3.3
郑州→南京→济南	是	租车自驾	高铁+高铁	8149	13
济南→南京→武汉	是	租车自驾	高铁+高铁	6257.4	8.9
武汉→南京→南昌	是	租车自驾	高铁+高铁	9320.1	12.1
南昌→南京→福州	是	租车自驾	高铁+高铁	7286.6	10.2
福州→南京→北京	是	租车自驾	飞机+高铁	8803.5	10.2
北京→西安	否	租车自驾	高铁	3744.9	4.2
西安→南京→太原	是	租车自驾	飞机+飞机	9214.8	4.9
太原→呼和浩特	否	租车自驾	飞机	3431.4	4.1
呼和浩特→南京→石家庄	是	租车自驾	飞机+飞机	7571.6	2.3

石家庄→南京→广州	是	租车自驾	飞机+飞机	9943	6.3
广州→南京→重庆	是	租车自驾	飞机+飞机	12574	8.4
重庆→南京→兰州	是	租车自驾	飞机+飞机	15076.8	8.9
兰州→西宁	否	租车自驾	高铁	4393	5.1
西宁→南京→沈阳	是	租车自驾	飞机+飞机	10561.8	2.5
沈阳→长春	否	租车自驾	高铁	2266.3	3.1
长春→南京→贵阳	是	租车自驾	飞机+飞机	10374.9	3.0
贵阳→海口	否	租车自驾	飞机	5596.1	3.6
海口→南京→南宁	是	租车自驾	飞机+飞机	12213.6	4.0
南宁→南京→银川	是	租车自驾	飞机+飞机	12140.8	3.9
银川→南京→哈尔滨	是	租车自驾	飞机+飞机	11490.7	3.5
哈尔滨→南京→昆明	是	租车自驾	飞机+飞机	17648.5	6.6
昆明→南京→拉萨	是	租车自驾	飞机+飞机	19578.4	6.8
拉萨→南京→成都	是	租车自驾	飞机+飞机	14403	1.7
成都→南京→乌鲁木齐	是	租车自驾	飞机+飞机	16312.8	10.5
乌鲁木齐→南京	是	租车自驾	飞机+飞机	11238.8	12
总和				277192.8	10年(换算)

根据从国家旅游局官网上收集的国家 5A 级旅游景区评定的相关信息，以及国家旅游局官网上收集的国家 4A 级景区名单，结合模型和算法分析，规划出了该旅游爱好者的十年旅游计划如上表 13 所示。从图中可以得出此次旅行的时间正好达到 10 年，且所使用的费用达到最低，为 27.72 万。下表 14 为优化后的行程安排。

表 14 行程安排

时间安排	路线安排
第一年	①南京→合肥 ②合肥→上海 ③上海→南京→杭州
第二年	①杭州→南京→长沙 ②长沙→天津 ③郑州→南京→济南
第三年	①郑州→南京→济南 ②济南→南京→武汉
第四年	①武汉→南京→南昌 ②南昌→南京→福州
第五年	①福州→南京→北京 ②北京→西安 ③西安→南京→太原 ④太原→呼和浩特
第六年	①呼和浩特→南京→石家庄 ②石家庄→南京→广州

	③广州→南京→重庆 ④重庆→南京→兰州
第七年	①兰州→西宁 ②西宁→南京→沈阳 ③沈阳→长春 ④长春→南京→贵阳
第八年	①贵阳→海口 ②海口→南京→南宁 ③南宁→南京→银川 ④银川→南京→哈尔滨
第九年	①哈尔滨→南京→昆明 ②昆明→南京→拉萨 ③拉萨→南京→成都 ④成都→南京→乌鲁木齐
第十年	①乌鲁木齐→南京

九 模型优缺点及推广

本文思路清晰，模型恰当，得出的方案合理；成功的使用了 0-1 变量，使得模型的建立和编程得意顺利进行；本模型需要搜集大量的数据来支持计算，对信息收集能力需要一定的要求。建立的模型步骤清晰，过程自然且流畅。随着问题的深入，本文不断地对模型加以改进、延伸，并通过参数约束，对模型结果进行分析，可知本文的模型精确度较高，且较为实用。

9.1 模型的评价

9.1.1 模型的优点

(1) 合理地引入 0-1 变量表示旅行者是否游玩过景点，得到将旅游时间转化为天数的分段函数表达式。

(2) 将旅游路途时间花销和游览时间花销转化为景点 i 到景点 j 的旅游天数时间花销，从而将旅游路线规划问题转化为 TSP 问题，简化求解。

(3) 在解题过程中，使用计算机进行编程，在分析和运算方面有较高的精度，时间大大缩短，使答案更加明了。

(4) 合理恰当的使用了表格和图形，使数据的体现和意思的表达更加清晰。

(5) 答案详细、具体，并且接近实际，具有较强的可操作性。

9.1.2 模型的缺点

(1) 采用蚁群算法和遗传算法。求解，不一定能保证得到全局最优解，且求解时间较长。

(2) 大多数数据来自于网络，数据缺乏可靠性。

(3) 假设旅游者只在早上出行，有失完备性。

9.2 模型的改进与推广

(1) 实际情况中，各地的旅游景点开放的时间段应该是不同的

(2) 如今网络很发达，可以通过网络买到价格很低的车票或是飞机票，而且不同时间段价格也不同，所以很难提前统计，要根据实际进行调整。

(3) 因数据资料搜集不完整，实际情况对变，准确性也有待商榷。

参考文献

- [1] 佟欣, 孙仲强, 徐斌. 黑龙江省旅游路线优化设计[J]. 齐齐哈尔大学学报(自然科学版), 2013, 03: 76-78+85.
- [2] 潘玉侠, 梁勤欧. 基于遗传算法的旅游线路优化[J]. 浙江师范大学学报(自然科学版), 2011, 03: 350-354.
- [3] JeKervliet, 11 fbrd Nowell (1999), HeteogeneousvisiotsrandhtesPatiallimitsofhtetrvaeleostmodel. Journal of Leisure Research, 31, 404 - 419.
- [4] Randall, A. (1994). A DiiffelytwihthterTaveleostMehtod. Land Economics, 70, 8896.
- [5] Smith, VK & KoPP, R.J. (1980). hTeSPatialLimitsofhterTavelCostRecreationalDemandModels. Land Economics, 56, 64 - 72.
- [6] 杨剑峰. 蚁群算法及其应用研究[D]. 浙江大学, 2007.
- [7] <http://map.baidu.com/>
- [8] 康钦容, 唐建新, 张卫中等. 改进模糊层次分析法在滑坡治理方案优化中的应用[J]. 重庆大学学报, 2010, 33(9): 98-103.

-附录-

目录

一 表格.....	- 37 -
表 1 第一问的结果.....	- 37 -
表 2 第二问的结果.....	- 45 -
表 3 第三问的结果.....	- 52 -
表 4 第三问的结果.....	- 62 -
二 代码.....	- 71 -
问题一代码.....	- 71 -
问题二代码.....	- 79 -
问题三代码.....	- 90 -
问题四代码.....	- 101 -

一 表格

表 1 第一问的结果

消耗时间 (天)	出发方式	出发地	费用 (元)	路途 时间 (h)	游览景区
0.5	租车自驾	西安市	304.6	0.3	西安大雁塔—大唐芙蓉园 景区
0.5	租车自驾	西安大雁塔—大唐芙蓉 园景区	322.6	0.3	西安华清池景区
0.5	租车自驾	西安华清池景区	305.5	0.2	西安秦始皇兵马俑博物馆
0.5	租车自驾	西安秦始皇兵马俑博物 馆	382	0.2	宝鸡扶风县法门寺佛文化 景区
1	租车自驾	宝鸡扶风县法门寺佛文 化景区	838.1	0.3	渭南华阴市华山风景区
1	租车自驾	渭南华阴市华山风景区	882	4	延安黄陵县黄帝陵景区
0.9	高铁	西安市	535.5	3	太原市
0.5	租车自驾	太原市	234.7	0.2	大同云冈石窟
1	租车自驾	大同云冈石窟	497.3	0.1	忻州五台山风景名胜區
0.5	租车自驾	忻州五台山风景名胜區	419.9	0.1	晋中市乔家大院文化园区
0.5	租车自驾	晋中市乔家大院文化园 区	171	0.2	晋中市平遥县平遥古城景 区
0.5	租车自驾	晋中市平遥县平遥古城 景区	282.9	0.3	晋中市介休市绵山风景名 胜区
0.5	租车自驾	晋中市介休市绵山风景 名胜区	415.6	0.3	晋城阳城县皇城相府生态 文化旅游区
0.6	飞机	太原市	1410	0.5	呼和浩特市
1	租车自驾	呼和浩特市	634.2	0.3	鄂尔多斯达拉特旗响沙湾

					旅游景区
0.5	租车自驾	鄂尔多斯达拉特旗响沙湾旅游景区	337.4	0.2	鄂尔多斯伊金霍洛旗成吉思汗陵旅游区
0.6	飞机	呼和浩特市	2130	0.7	天津市
0.5	租车自驾	天津市	152.8	0.1	天津古文化街旅游区(津门故里)
1	租车自驾	天津古文化街旅游区(津门故里)	424.4	0.2	天津蓟县盘山风景名胜
0.2	飞机	天津市	3240	1.2	西安市
0.3	高铁	西安市	687	2.4	郑州市
0.5	租车自驾	郑州市	293.9	0.3	开封清明上河园景区
0.5	租车自驾	开封清明上河园景区	393.1	0.2	安阳殷墟景区
1	租车自驾	安阳殷墟景区	736.3	4	洛阳龙门石窟景区
1	租车自驾	洛阳龙门石窟景区	428.8	0.3	平顶山鲁山县尧山—中原大佛景区
0.5	租车自驾	平顶山鲁山县尧山—中原大佛景区	384.1	0.1	洛阳嵩县白云山景区
1	租车自驾	洛阳嵩县白云山景区	651.7	0.1	郑州登封嵩山少林景区
1	租车自驾	郑州登封嵩山少林景区	732.6	0.1	洛阳新安县龙潭大峡谷景区
1	租车自驾	洛阳新安县龙潭大峡谷景区	816.1	0.1	洛阳栾川县老君山—鸡冠洞旅游区
1	租车自驾	洛阳栾川县老君山—鸡冠洞旅游区	719	0.3	南阳西峡伏牛山老界岭·恐龙遗址园旅游区
2.5	租车自驾	南阳西峡伏牛山老界岭·恐龙遗址园旅游区	1511.4	4	焦作(云台山—神农山—青天河)风景区
0.3	高铁	郑州市	687	2.4	西安市
1.1	高铁	西安市	1225.5	4.5	石家庄市
1	租车自驾	石家庄市	382.2	0.1	石家庄平山县西柏坡景区
1	租车自驾	石家庄平山县西柏坡景区	533.4	0.2	保定安新白洋淀景区(文化苑-大观园-鸳鸯岛-元妃荷园-嘎子印象-渔人乐园)
1	租车自驾	保定安新白洋淀景区(文化苑-大观园-鸳鸯岛-元妃荷园-嘎子印象-渔人乐园)	613.3	0.2	保定涞水县野三坡景区(百里峡-白草畔-鱼谷洞-龙门天关)
1.5	租车自驾	保定涞水县野三坡景区(百里峡-白草畔-鱼谷洞-龙门天关)	1004.4	4	承德避暑山庄及周围寺庙景区(普陀宗乘-须弥福寺-普宁寺-普佑寺)
1	租车自驾	承德避暑山庄及周围寺庙景区(普陀宗乘-须弥福寺-普宁寺-普佑寺)	689.7	0.1	秦皇岛山海关景区(老龙头-山海关古城-天下第一关-孟姜女庙)
0.1	飞机	石家庄市	2880	1	银川市
0.5	租车自驾	银川市	315	0.3	银川市灵武水洞沟旅游区
0.5	租车自驾	银川市灵武水洞沟旅游区	356.9	0.3	银川镇北堡西部影视城
1	租车自驾	银川镇北堡西部影视城	354.4	0.3	石嘴山平罗县沙湖旅游景区
0.5	租车自驾	石嘴山平罗县沙湖旅游景区	564.4	0.3	中卫沙坡头旅游景区
0.1	飞机	银川市	2550	0.8	西安市

0.5	高铁	西安市	1363.5	3.9	长沙市
0.5	租车自驾	长沙市	193.3	0.1	长沙市宁乡县花明楼景区
0.5	租车自驾	长沙市宁乡县花明楼景区	270.9	0.3	长沙岳麓山—橘子洲旅游区
0.5	租车自驾	长沙岳麓山—橘子洲旅游区	359.9	0.1	衡阳南岳衡山旅游区
0.5	租车自驾	衡阳南岳衡山旅游区	416.2	0.1	郴州市东江湖旅游区
1	租车自驾	郴州市东江湖旅游区	920.1	4	湘潭韶山旅游区
0.5	租车自驾	湘潭韶山旅游区	478.7	0.2	岳阳岳阳楼—君山岛景区
2.5	租车自驾	岳阳岳阳楼—君山岛景区	1452	4	张家界武陵源—天门山旅游区
0.1	飞机	长沙市	4260	0.9	上海市
0.5	租车自驾	上海市	303.5	0.3	东方明珠广播电视塔
0.5	租车自驾	东方明珠广播电视塔	302.1	0.2	上海科技馆
1	租车自驾	上海科技馆	603.1	0.2	上海野生动物园
0.2	飞机	上海市	4170	1.5	西安市
1.2	高铁	西安市	1546.5	5.6	北京市
0.5	租车自驾	北京市	301	0.1	故宫博物院
0.5	租车自驾	故宫博物院	302.9	0.3	恭王府景区
0.5	租车自驾	恭王府景区	305.1	0.1	天坛公园
0.5	租车自驾	天坛公园	311.9	0.3	北京奥林匹克公园(鸟巢-水立方-中国科技馆-国家奥林匹克森林公园)
0.5	租车自驾	北京奥林匹克公园(鸟巢-水立方-中国科技馆-国家奥林匹克森林公园)	307.7	0.1	颐和园
0.5	租车自驾	颐和园	321.5	0.1	明十三陵景区(神路-定陵-长陵-昭陵)
0.5	租车自驾	明十三陵景区(神路-定陵-长陵-昭陵)	348.3	0.3	八达岭—慕田峪长城旅游区
0.6	飞机	北京市	3150	1.1	长春市
1	租车自驾	长春市	306.2	0.1	长白山景区
0.5	租车自驾	长白山景区	302.2	0.2	长春净月潭景区
0.5	租车自驾	长春净月潭景区	303.5	0.3	长春市长影世纪城景区
0.5	租车自驾	长春市长影世纪城景区	313	0.2	长春伪满皇宫博物馆
0.3	飞机	长春市	4890	2.3	西安市
0.7	高铁	西安市	1761	5.3	武汉市
0.5	租车自驾	武汉市	230.8	0.3	武汉黄鹤楼公园
0.5	租车自驾	武汉黄鹤楼公园	243.8	0.3	武汉市东湖景区
0.5	租车自驾	武汉市东湖景区	403.6	0.2	武汉市黄陂木兰文化生态旅游区
1	租车自驾	武汉市黄陂木兰文化生态旅游区	731	4	宜昌秭归县屈原故里文化旅游区
0.5	租车自驾	宜昌秭归县屈原故里文化旅游区	234.5	0.1	宜昌三峡大坝旅游区
0.5	租车自驾	宜昌三峡大坝旅游区	398.1	0.1	宜昌长阳县清江画廊景区
0.5	租车自驾	宜昌长阳县清江画廊景区	534.2	0.2	恩施土家族苗族自治州巴东神龙溪纤夫文化旅游区
1	租车自驾	恩施土家族苗族自治州巴东神龙溪纤夫文化旅游	640.9	0.3	宜昌三峡人家风景区

		区			
1.5	租车自驾	宜昌三峡人家风景区	1104.8	4	十堰丹江口市武当山风景区
2	租车自驾	十堰丹江口市武当山风景区	1392.7	0.1	神农架生态旅游区
1.5	租车自驾	神农架生态旅游区	1304.2	4	恩施土家族苗族自治州恩施大峡谷景区
0.7	高铁	武汉市	1761	5.3	西安市
0.1	飞机	西安市	2130	1	西宁市
0.5	租车自驾	西宁市	165.3	0.2	西宁市湟中县塔尔寺景区
1	租车自驾	西宁市湟中县塔尔寺景区	616.5	0.2	青海湖风景区
0	高铁	西宁市	0	0	海口市
0.5	租车自驾	海口市	480.6	0.2	保亭县呀诺达雨林文化旅游区
0.5	租车自驾	保亭县呀诺达雨林文化旅游区	230.2	0.3	保亭县海南槟榔谷黎苗文化旅游区
0.5	租车自驾	保亭县海南槟榔谷黎苗文化旅游区	287.1	0.3	三亚南山文化旅游区
0.5	租车自驾	三亚南山文化旅游区	230.6	0.1	三亚南山大小洞天旅游区
0.5	租车自驾	三亚南山大小洞天旅游区	425.1	0.1	陵水县分界洲岛旅游区
0.3	飞机	海口市	5490	2.3	西安市
0.1	飞机	西安市	2220	0.8	成都市
0.5	租车自驾	成都市	368	0.2	乐山乐山大佛景区
1	租车自驾	乐山乐山大佛景区	651.1	0.2	成都青城山—都江堰旅游景区
1	租车自驾	成都青城山—都江堰旅游景区	608	0.3	绵阳北川羌城旅游区(中国羌城—老县城地震遗址—“5·12”特大地震纪念馆—北川羌族民俗博物馆—北川新县城—吉娜羌寨)
1	租车自驾	绵阳北川羌城旅游区(中国羌城—老县城地震遗址—“5·12”特大地震纪念馆—北川羌族民俗博物馆—北川新县城—吉娜羌寨)	459.4	0.2	广元市剑门蜀道剑门关旅游景区
0.5	租车自驾	广元市剑门蜀道剑门关旅游景区	377.6	0.2	南充市阆中古城旅游景区
0.5	租车自驾	南充市阆中古城旅游景区	417.5	0.3	广安市邓小平故里旅游区
1.5	租车自驾	广安市邓小平故里旅游区	1302.1	4	阿坝藏族羌族自治州汶川特别旅游区(震中映秀—水磨古镇—三江生态旅游区)
1	租车自驾	阿坝藏族羌族自治州汶川特别旅游区(震中映秀—水磨古镇—三江生态旅游区)	913.6	4	阿坝藏族羌族自治州松潘县黄龙风景名胜景区
1	租车自驾	阿坝藏族羌族自治州松	727.1	0.1	阿坝藏族羌族自治州九寨沟

		潘县黄龙风景名胜区			沟景区
2	租车自驾	阿坝藏族羌族自治州九寨沟景区	1818.4	8	乐山峨眉山景区
0.1	飞机	成都市	2220	0.8	西安市
0.1	飞机	西安市	2880	1	济南市
0.5	租车自驾	济南市	361	0.3	济宁曲阜明故城三孔旅游区
0.5	租车自驾	济宁曲阜明故城三孔旅游区	451.3	0.1	枣庄台儿庄古城景区
1	租车自驾	枣庄台儿庄古城景区	667	0.2	泰安泰山景区
1	租车自驾	泰安泰山景区	532.2	0.3	济南天下第一泉景区(趵突泉—大明湖—五龙潭—环城公园-黑虎泉)
1	租车自驾	济南天下第一泉景区(趵突泉—大明湖—五龙潭—环城公园-黑虎泉)	799.7	0.1	山东沂蒙山旅游区(沂山景区—龟蒙景区—云蒙景区)
0.5	租车自驾	山东沂蒙山旅游区(沂山景区—龟蒙景区—云蒙景区)	428.4	0.1	青岛崂山景区
0.5	租车自驾	青岛崂山景区	440	0.1	烟台蓬莱阁—三仙山—八仙过海旅游区
0.5	租车自驾	烟台蓬莱阁—三仙山—八仙过海旅游区	348.7	0.3	烟台龙口南山景区
0.5	租车自驾	烟台龙口南山景区	397.6	0.1	威海刘公岛景区
0.1	飞机	济南市	2880	1	西安市
0.1	飞机	西安市	3000	1.2	贵阳市
0.5	租车自驾	贵阳市	338.5	0.3	安顺龙宫景区
0.5	租车自驾	安顺龙宫景区	273.9	0.1	安顺镇宁县黄果树瀑布景区
1	租车自驾	安顺镇宁县黄果树瀑布景区	841.1	0.2	毕节市百里杜鹃景区
1.5	租车自驾	毕节市百里杜鹃景区	1352.6	4	黔南布依族苗族自治州荔波樟江景区
0.1	飞机	贵阳市	2490	0.6	南宁市
0.5	租车自驾	南宁市	302.9	0.1	南宁市青秀山旅游区
1	租车自驾	南宁市青秀山旅游区	838.8	4	桂林独秀峰·靖江王城景区
1	租车自驾	桂林独秀峰·靖江王城景区	614.7	0.3	桂林漓江风景区
1	租车自驾	桂林漓江风景区	694.4	0.2	桂林兴安县乐满地度假世界
0.2	飞机	南宁市	5400	1.8	西安市
0.1	飞机	西安市	3000	0.7	兰州市
1	租车自驾	兰州市	776.6	4	天水麦积山景区
1	租车自驾	天水麦积山景区	673.4	0.3	平凉崆峒山风景名胜区
1.5	租车自驾	平凉崆峒山风景名胜区	1726.9	8	嘉峪关文物景区
1.5	租车自驾	嘉峪关文物景区	1042.1	4	酒泉市敦煌沙山月牙泉景区
0.1	飞机	兰州市	3000	0.7	西安市
0.2	飞机	西安市	3030	1.2	南昌市
0.5	租车自驾	南昌市	407.8	0.3	景德镇古窑民俗博览区

1	租车自驾	景德镇古窑民俗博览区	400.1	0.2	上饶婺源县江湾景区
1	租车自驾	上饶婺源县江湾景区	416.7	0.1	上饶三清山旅游景区
1	租车自驾	上饶三清山旅游景区	634.7	0.1	鹰潭市贵溪龙虎山风景名胜 区
1	租车自驾	鹰潭市贵溪龙虎山风景 名胜区	751.8	4	赣州市瑞金市共和国摇篮 景区
1.5	租车自驾	赣州市瑞金市共和国摇 篮景区	987.5	4	吉安井冈山风景旅游区
2	租车自驾	吉安井冈山风景旅游区	1351	8	九江庐山风景名胜 区
0.2	飞机	南昌市	3030	1.2	西安市
0.1	飞机	西安市	3180	1	合肥市
0.5	租车自驾	合肥市	320.2	0.3	阜阳市颍上县八里河风景 区
1.5	租车自驾	阜阳市颍上县八里河风 景区	721.4	4	安庆潜山县天柱山风景区
1	租车自驾	安庆潜山县天柱山风景 区	476.1	0.2	池州青阳县九华山风景区
0.5	租车自驾	池州青阳县九华山风景 区	368.7	0.3	黄山市古徽州文化旅游区 (徽州古城—牌坊群鲍家花 园—唐模—潜口民宅—呈 坎)
1	租车自驾	黄山市古徽州文化旅 游区(徽州古城—牌坊群鲍 家花园—唐模—潜口民宅 —呈坎)	353.4	0.2	黄山市黟县皖南古村落— 西递宏村
1	租车自驾	黄山市黟县皖南古村落 —西递宏村	402.6	0.3	宣城市绩溪县龙川景区
1	租车自驾	宣城市绩溪县龙川景区	544.8	0.1	黄山市黄山风景区
1.5	租车自驾	黄山市黄山风景区	848.8	4	六安市金寨县天堂寨旅游 景区
0.1	飞机	合肥市	3180	1	西安市
0.1	飞机	西安市	3600	0.8	重庆市
0.5	租车自驾	重庆市	396.1	0.1	大足石刻景区
1	租车自驾	大足石刻景区	800.3	0.3	南川金佛山—神龙峡风景 区
1	租车自驾	南川金佛山—神龙峡风 景区	652.1	0.1	万盛黑山谷-龙鳞石海风景 区
1.5	租车自驾	万盛黑山谷-龙鳞石海 风景区	1226.5	4	酉阳桃花源旅游景区
1.5	租车自驾	酉阳桃花源旅游景区	1424.2	8	巫山小三峡—小小三峡旅 游区
3	租车自驾	巫山小三峡—小小三峡 旅游区	2387.6	8	武隆喀斯特旅游区(天生三 桥、仙女山、芙蓉洞)
0.1	飞机	重庆市	3600	0.8	西安市
0.2	飞机	西安市	3840	1.5	杭州市
0.5	租车自驾	杭州市	289.3	0.3	绍兴市鲁迅故里—沈园景 区
0.5	租车自驾	绍兴市鲁迅故里—沈园 景区	331.4	0.2	嘉兴南湖旅游区
0.5	租车自驾	嘉兴南湖旅游区	247.4	0.1	嘉兴桐乡乌镇古镇旅游 区

0.5	租车自驾	嘉兴桐乡乌镇古镇旅游区	237.7	0.1	湖州市南浔区南浔古镇景区
0.5	租车自驾	湖州市南浔区南浔古镇景区	398.1	0.1	杭州西溪湿地旅游区
0.5	租车自驾	杭州西溪湿地旅游区	300	0.1	杭州西湖风景区
0.5	租车自驾	杭州西湖风景区	371.1	0.3	金华东阳横店影视城景区
0.5	租车自驾	金华东阳横店影视城景区	371.1	0.3	宁波奉化溪口—滕头旅游景区
1	租车自驾	宁波奉化溪口—滕头旅游景区	602	0.1	舟山普陀山风景区
1.5	租车自驾	舟山普陀山风景区	1001.5	4	温州乐清市雁荡山风景区
1	租车自驾	温州乐清市雁荡山风景区	845	4	衢州市开化根宫佛国文化旅游区
1	租车自驾	衢州市开化根宫佛国文化旅游区	784.3	0.3	杭州淳安千岛湖风景区
0.2	飞机	杭州市	3840	1.5	西安市
0.7	飞机	西安市	4500	1.9	沈阳市
0.5	租车自驾	沈阳市	303.4	0.3	沈阳植物园
1	租车自驾	沈阳植物园	401.9	0.3	本溪市本溪水洞景区
1	租车自驾	本溪市本溪水洞景区	868.9	4	大连金石滩景区(地质公园-发现王国-蜡像馆-文化博览广场)
0.5	租车自驾	大连金石滩景区(地质公园-发现王国-蜡像馆-文化博览广场)	282.6	0.2	大连老虎滩海洋公园—老虎滩极地馆
0.2	飞机	沈阳市	4500	1.9	西安市
0.2	飞机	西安市	4590	1.8	福州市
0.5	租车自驾	福州市	226.6	0.1	福州市三坊七巷景区
0.5	租车自驾	福州市三坊七巷景区	486.8	0.1	厦门鼓浪屿风景名胜区
1	租车自驾	厦门鼓浪屿风景名胜区	541.5	0.3	泉州市清源山风景名胜区
1	租车自驾	泉州市清源山风景名胜区	788.1	0.1	福建土楼（永定·南靖）旅游景区
2	租车自驾	福建土楼（永定·南靖）旅游景区	1452.6	8	南平武夷山风景名胜区
1	租车自驾	南平武夷山风景名胜区	759.2	0.1	三明泰宁风景旅游区
1.5	租车自驾	三明泰宁风景旅游区	1183.6	4	宁德屏南（白水洋·鸳鸯溪）旅游景区
1	租车自驾	宁德屏南（白水洋·鸳鸯溪）旅游景区	804.6	0.1	宁德市福鼎太姥山旅游区
0.2	飞机	福州市	4590	1.8	西安市
0.2	飞机	西安市	4620	1.8	昆明市
0.5	租车自驾	昆明市	309	0.2	昆明石林风景区
1	租车自驾	昆明石林风景区	868.9	4	大理崇圣寺三塔文化旅游区
0.5	租车自驾	大理崇圣寺三塔文化旅游区	394.8	0.1	丽江古城景区
1	租车自驾	丽江古城景区	473.2	0.3	丽江玉龙雪山景区
1	租车自驾	丽江玉龙雪山景区	528.5	0.1	迪庆藏族自治州香格里拉普达措国家公园
2	租车自驾	迪庆藏族自治州香格里拉	1764	8	中科院西双版纳热带植物

		拉普达措国家公园			园
0.2	飞机	昆明市	4620	1.8	西安市
0.2	飞机	西安市	4620	1.8	广州市
0.5	租车自驾	广州市	159.7	0.3	佛山市德顺区长鹿旅游休 博园
0.5	租车自驾	佛山市德顺区长鹿旅游 休博园	239	0.3	佛山西樵山景区
0.5	租车自驾	佛山西樵山景区	280.8	0.1	广州白云山景区
0.5	租车自驾	广州白云山景区	349.1	0.2	深圳观澜湖休闲旅游区
0.5	租车自驾	深圳观澜湖休闲旅游区	244.8	0.3	深圳华侨城旅游度假区
1	租车自驾	深圳华侨城旅游度假区	420.4	0.2	惠州市罗浮山景区
1	租车自驾	惠州市罗浮山景区	555.8	0.3	广州长隆旅游度假区
1	租车自驾	广州长隆旅游度假区	580.9	4	韶关仁化丹霞山景区
0.5	租车自驾	韶关仁化丹霞山景区	478.7	0.2	清远连州地下河旅游景区
1.5	租车自驾	清远连州地下河旅游景 区	1344.8	8	梅州市梅县区雁南飞茶田 景区
0.2	飞机	广州市	4620	1.8	西安市
0.3	飞机	西安市	5520	2.6	哈尔滨市
0.5	租车自驾	哈尔滨市	306.1	0.1	哈尔滨太阳岛景区
1	租车自驾	哈尔滨太阳岛景区	818	4	黑河五大连池景区
1	租车自驾	黑河五大连池景区	1000.4	4	伊春市汤旺河林海奇石景 区
1.5	租车自驾	伊春市汤旺河林海奇石 景区	1366.8	8	牡丹江宁安市镜泊湖景区
2	租车自驾	牡丹江宁安市镜泊湖景 区	2877.2	8	大兴安岭地区漠河北极村 旅游景区
0.3	飞机	哈尔滨市	5520	2.6	西安市
0.4	飞机	西安市	5970	3.1	拉萨市
0.5	租车自驾	拉萨市	302.1	0.2	拉萨大昭寺景区
0.5	租车自驾	拉萨大昭寺景区	300.9	0.2	拉萨布达拉宫景区
0.4	飞机	拉萨市	5970	3.1	西安市
0.4	飞机	西安市	6600	2.8	乌鲁木齐市
0.5	租车自驾	乌鲁木齐市	489.1	0.2	吐鲁番葡萄沟风景区
1	租车自驾	吐鲁番葡萄沟风景区	657.1	0.1	乌鲁木齐天山大峡谷
1	租车自驾	乌鲁木齐天山大峡谷	595.7	0.3	昌吉州阜康市天山天池风 景名胜区
1.5	租车自驾	昌吉州阜康市天山天池 风景名胜区	1431.4	4	巴音郭楞蒙古自治州博湖 县博斯腾湖景区
1.5	租车自驾	巴音郭楞蒙古自治州博 湖县博斯腾湖景区	1256.7	4	伊犁地区新源县那拉提旅 游风景区
2	租车自驾	伊犁地区新源县那拉提 旅游风景区	1672.2	8	喀什地区泽普县金胡杨景 区
0.5	租车自驾	喀什地区泽普县金胡杨 景区	443.3	0.2	喀什地区噶尔老城景区
2	租车自驾	喀什地区噶尔老城景区	3042.5	8	阿勒泰地区布尔津县喀纳 斯景区
2	租车自驾	阿勒泰地区布尔津县喀 纳斯景区	1650.8	8	阿勒泰地区富蕴县可可托 海景区
0.2	飞机	西安市	3540	1.2	南京市
0.5	租车自驾	南京市	307.2	0.1	南京夫子庙—秦淮河风光

					带(江南贡院-白鹭洲-中华门-瞻园-王谢故居)
0.5	租车自驾	南京夫子庙—秦淮风光带(江南贡院-白鹭洲-中华门-瞻园-王谢故居)	304.3	0.2	南京钟山—中山陵风景名胜区(明孝陵-音乐台-灵谷寺-梅花山-紫金山天文台)
0.5	租车自驾	南京钟山—中山陵风景名胜区(明孝陵-音乐台-灵谷寺-梅花山-紫金山天文台)	321.7	0.3	扬州瘦西湖风景区
0.5	租车自驾	扬州瘦西湖风景区	306.3	0.1	泰州姜堰区溱湖国家湿地公园
0.5	租车自驾	泰州姜堰区溱湖国家湿地公园	245.3	0.1	苏州昆山周庄古镇景区
0.5	租车自驾	苏州昆山周庄古镇景区	269.3	0.3	苏州园林(拙政园—留园—虎丘)
0.5	租车自驾	苏州园林(拙政园—留园—虎丘)	231.7	0.3	苏州市金鸡湖国家商务旅游示范区
0.5	租车自驾	苏州市金鸡湖国家商务旅游示范区	240.8	0.2	苏州吴江同里古镇景区
0.5	租车自驾	苏州吴江同里古镇景区	308.3	0.3	无锡鼋头渚景区
0.5	租车自驾	无锡鼋头渚景区	244.1	0.3	无锡灵山大佛景区
0.5	租车自驾	无锡灵山大佛景区	304.9	0.1	常州环球恐龙城景区(中华恐龙园-恐龙谷温泉-恐龙城大剧院)
0.5	租车自驾	常州环球恐龙城景区(中华恐龙园-恐龙谷温泉-恐龙城大剧院)	354.6	0.3	中央电视台无锡影视基地三国水浒城景区
0.5	租车自驾	中央电视台无锡影视基地三国水浒城景区	325.4	0.2	南通市濠河风景区
0.5	租车自驾	南通市濠河风景区	495.5	0.3	淮安市周恩来故里景区(周恩来纪念馆-周恩来故居-附马巷历史街区-河下古镇)
1	租车自驾	淮安市周恩来故里景区(周恩来纪念馆-周恩来故居-附马巷历史街区-河下古镇)	513.5	0.2	镇江三山风景名胜区(金山—北固山—焦山)
1	租车自驾	镇江三山风景名胜区(金山—北固山—焦山)	537.9	0.1	镇江句容茅山景区
1	租车自驾	镇江句容茅山景区	536.5	0.3	常州溧阳市天目湖景区(天目湖-南山竹海-御水温泉)
1	租车自驾	常州溧阳市天目湖景区(天目湖-南山竹海-御水温泉)	596	0.1	苏州常熟沙家浜—虞山尚湖旅游区
1	租车自驾	苏州常熟沙家浜—虞山尚湖旅游区	521.6	0.3	苏州吴中太湖旅游区(旺山—穹窿山—东山)

表 2 问题 2 的数据

消耗	出发地	行车时	行车里	游览景区
----	-----	-----	-----	------

时间(天)		间(h)	程(公里)	
0.5	西安市	0.2	7.7	西安大雁塔—大唐芙蓉园景区
0.5	西安大雁塔—大唐芙蓉园景区	0.5	44	西安秦始皇兵马俑博物馆
0.5	西安秦始皇兵马俑博物馆	0.2	9.1	西安华清池景区
0.5	西安华清池景区	1.7	151.7	宝鸡扶风县法门寺佛文化景区
1	宝鸡扶风县法门寺佛文化景区	2.5	227.4	渭南华阴市华山风景区
1	渭南华阴市华山风景区	3.1	282	延安黄陵县黄帝陵景区
1.2	西安市	9.8	880	西宁市
0.5	西宁市	0.6	25.5	西宁市湟中县塔尔寺景区
1	西宁市湟中县塔尔寺景区	0.7	27.5	青海湖风景区
0.9	西宁市	7.1	640	银川市
0.5	银川市	0.6	25	银川市灵武水洞沟旅游区
0.5	银川市灵武水洞沟旅游区	0.6	56.9	银川镇北堡西部影视城
0.5	银川镇北堡西部影视城	2.8	251.6	中卫沙坡头旅游景区
1	中卫沙坡头旅游景区	2.7	241.9	石嘴山平罗县沙湖旅游景区
0.9	银川市	7.4	670	呼和浩特市
1	呼和浩特市	2	184.2	鄂尔多斯达拉特旗响沙湾旅游景区
0.5	鄂尔多斯达拉特旗响沙湾旅游景区	1.2	112.4	鄂尔多斯伊金霍洛旗成吉思汗陵旅游区
1.3	呼和浩特市	10.8	970	西安市
1.5	西安市	12.2	1100	天津市
0.5	天津市	0.1	4.6	天津古文化街旅游区(津门故里)
1	天津古文化街旅游区(津门故里)	1.4	124.4	天津蓟县盘山风景名胜区
1.5	天津市	12.2	1100	上海市
0.5	上海市	0.1	5.9	东方明珠广播电视塔
0.5	东方明珠广播电视塔	0.1	3.5	上海科技馆
1	上海科技馆	0.1	5.1	上海野生动物园
1.9	上海市	15.4	1390	西安市
3.9	西安市	31.4	2830	拉萨市
0.5	拉萨市	0.1	3.5	拉萨大昭寺景区
0.5	拉萨大昭寺景区	0	1.5	拉萨布达拉宫景区
3.9	拉萨市	31.4	2830	西安市
1.5	西安市	11.9	1070	贵阳市
0.5	贵阳市	1.3	113.5	安顺龙宫景区
0.5	安顺龙宫景区	0.5	48.9	安顺镇宁县黄果树瀑布景区
1	安顺镇宁县黄果树瀑布景区	2.7	241.1	毕节市百里杜鹃景区
1.5	毕节市百里杜鹃景区	4.3	384.9	黔南布依族苗族自治州荔波樟江景区
1.4	贵阳市	11.6	1040	海口市
0.5	海口市	2	183.7	陵水县分界洲岛旅游区
0.5	陵水县分界洲岛旅游区	0.9	82.1	保亭县海南槟榔谷黎苗文化旅游区
0.5	保亭县海南槟榔谷黎苗文化旅游区	0.2	8.7	保亭县呀诺达雨林文化旅游区
0.5	保亭县呀诺达雨林文化旅游区	0.8	70.5	三亚南山文化旅游区
0.5	三亚南山文化旅游区	0.2	9.3	三亚南山大小洞天旅游区
2.9	海口市	23.3	2100	西安市
1.5	西安市	12	1080	北京市
0.5	北京市	0	1.6	故宫博物院

0.5	故宫博物院	0.1	4.9	恭王府景区
0.5	恭王府景区	0.2	8.5	天坛公园
0.5	天坛公园	0.4	17.5	北京奥林匹克公园(鸟巢-水立方-中国科技馆-国家奥林匹克森林公园)
0.5	北京奥林匹克公园(鸟巢-水立方-中国科技馆-国家奥林匹克森林公园)	0.3	12.9	颐和园
0.5	颐和园	0.8	75.9	八达岭—慕田峪长城旅游区
0.5	八达岭—慕田峪长城旅游区	0.5	48.3	明十三陵景区(神路-定陵-长陵-昭陵)
1.4	北京市	10.9	980	长春市
0.5	长春市	0.3	11.6	长春伪满皇宫博物馆
0.5	长春伪满皇宫博物馆	0.2	8.1	长春净月潭景区
0.5	长春净月潭景区	0.1	5.8	长春市长影世纪城景区
1	长春市长影世纪城景区	0.1	5.2	长白山景区
2.8	长春市	22.7	2040	西安市
0.8	西安市	6.7	600	太原市
0.5	太原市	0.4	16.2	大同云冈石窟
1	大同云冈石窟	2.2	197.3	忻州五台山风景名胜区
0.5	忻州五台山风景名胜区	3	269.9	晋中市乔家大院文化园区
0.5	晋中市乔家大院文化园区	0.9	35	晋中市平遥县平遥古城景区
0.5	晋中市平遥县平遥古城景区	0.6	57.9	晋中市介休市绵山风景名胜区
0.5	晋中市介休市绵山风景名胜区	3	265.6	晋城阳城县皇城相府生态文化旅游区
1.7	太原市	13.2	1190	沈阳市
0.5	沈阳市	0.1	5.6	沈阳植物园
1	沈阳植物园	1.1	101.9	本溪市本溪水洞景区
1	本溪市本溪水洞景区	4.7	418.9	大连金石滩景区(地质公园-发现王国-蜡像馆-文化博览广场)
0.5	大连金石滩景区(地质公园-发现王国-蜡像馆-文化博览广场)	0.6	57.6	大连老虎滩海洋公园—老虎滩极地馆
2.4	沈阳市	19.4	1750	西安市
2.3	西安市	18.2	1640	南宁市
0.5	南宁市	0.1	4.8	南宁市青秀山旅游区
1	南宁市青秀山旅游区	4.3	388.8	桂林独秀峰·靖江王城景区
1	桂林独秀峰·靖江王城景区	0.6	24.5	桂林漓江风景区
1	桂林漓江风景区	1	94.4	桂林兴安县乐满地度假世界
2.3	南宁市	18.2	1640	西安市
0.9	西安市	7.2	650	兰州市
1	兰州市	3.6	326.6	天水麦积山景区
1	天水麦积山景区	2.5	223.4	平凉崆峒山风景名胜区
1.5	平凉崆峒山风景名胜区	11.7	1051.9	嘉峪关文物景区
1.5	嘉峪关文物景区	4.1	367.1	酒泉市敦煌沙山月牙泉景区
0.9	兰州市	7.2	650	西安市
1	西安市	7.8	700	重庆市
0.5	重庆市	1.1	96.1	大足石刻景区
1	大足石刻景区	2.2	200.3	南川金佛山—神龙峡风景区
0.5	南川金佛山—神龙峡风景区	1.5	139	武隆喀斯特旅游区(天生三硚、仙女山、芙蓉洞)
1	武隆喀斯特旅游区(天生三硚、仙女山、芙蓉洞)	1.8	163.7	万盛黑山谷-龙鳞石海风景区

1.5	万盛黑山谷-龙鳞石海风景区	3.6	326.5	酉阳桃花源旅游景区
1.5	酉阳桃花源旅游景区	5.8	524.2	巫山小三峡—小小三峡旅游区
1	重庆市	7.8	700	西安市
1.1	西安市	8.8	790	石家庄市
1	石家庄市	0.9	82.2	石家庄平山县西柏坡景区
1	石家庄平山县西柏坡景区	2.6	233.4	保定安新白洋淀景区(文化苑-大观园-鸳鸯岛-元妃荷园-嘎子印象-渔人乐园)
1	保定安新白洋淀景区(文化苑-大观园-鸳鸯岛-元妃荷园-嘎子印象-渔人乐园)	1.8	163.3	保定涞水县野三坡景区(百里峡-白草畔-鱼谷洞-龙门天关)
1.5	保定涞水县野三坡景区(百里峡-白草畔-鱼谷洞-龙门天关)	3.7	329.4	承德避暑山庄及周围寺庙景区(普陀宗乘-须弥福寺-普宁寺-普佑寺)
1	承德避暑山庄及周围寺庙景区(普陀宗乘-须弥福寺-普宁寺-普佑寺)	2.7	239.7	秦皇岛山海关景区(老龙头-山海关古城-天下第一关-孟姜女庙)
1.1	石家庄市	8.8	790	西安市
1	西安市	8.2	740	长沙市
0.5	长沙市	0.2	8.1	长沙岳麓山—橘子洲旅游区
0.5	长沙岳麓山—橘子洲旅游区	0.5	45.9	长沙市宁乡县花明楼景区
0.5	长沙市宁乡县花明楼景区	0.6	25.8	湘潭韶山旅游区
0.5	湘潭韶山旅游区	1.5	132.1	衡阳南岳衡山旅游区
0.5	衡阳南岳衡山旅游区	2.1	191.2	郴州市东江湖旅游区
1.5	郴州市东江湖旅游区	5	450.2	岳阳岳阳楼—君山岛景区
2.5	岳阳岳阳楼—君山岛景区	3.6	327	张家界武陵源—天门山旅游区
1	长沙市	8.2	740	西安市
1.3	西安市	10.1	910	济南市
0.5	济南市	1.5	136	济宁曲阜明故城三孔旅游区
0.5	济宁曲阜明故城三孔旅游区	1.7	151.3	枣庄台儿庄古城景区
1	枣庄台儿庄古城景区	2.4	217	泰安泰山景区
1	泰安泰山景区	0.8	72.7	济南天下第一泉景区(趵突泉—大明湖—五龙潭—环城公园-黑虎泉)
1	济南天下第一泉景区(趵突泉—大明湖—五龙潭—环城公园-黑虎泉)	2.1	191.9	山东沂蒙山旅游区(沂山景区—龟蒙景区—云蒙景区)
0.5	山东沂蒙山旅游区(沂山景区—龟蒙景区—云蒙景区)	2.3	203.4	青岛崂山景区
0.5	青岛崂山景区	2.1	190.1	烟台龙口南山景区
0.5	烟台龙口南山景区	0.5	48.7	烟台蓬莱阁—三仙山—八仙过海旅游区
0.5	烟台蓬莱阁—三仙山—八仙过海旅游区	1.7	152.3	威海刘公岛景区
1.3	济南市	10.1	910	西安市
1.3	西安市	10.2	920	合肥市
0.5	合肥市	1.9	170.2	阜阳市颍上县八里河风景区
1	阜阳市颍上县八里河风景区	4.9	442.8	黄山市古徽州文化旅游区(徽州古城—牌坊群鲍家花园—唐模—潜口民宅—呈坎)
1	黄山市古徽州文化旅游区(徽州古城—牌坊群鲍家花园—唐模—潜口民宅—呈坎)	0.6	53.4	黄山市黟县皖南古村落—西递宏村

1	黄山市黟县皖南古村落—西递宏村	0.5	42.4	黄山市黄山风景区
1	黄山市黄山风景区	1.1	94.8	宣城市绩溪县龙川景区
1	宣城市绩溪县龙川景区	2.1	191.5	池州青阳县九华山风景区
1	池州青阳县九华山风景区	2	176.1	安庆潜山县天柱山风景区
1	安庆潜山县天柱山风景区	2.2	199.2	六安市金寨县天堂寨旅游景区
1.3	合肥市	10.2	920	西安市
2.2	西安市	17.7	1590	昆明市
0.5	昆明市	0.9	84	昆明石林风景区
1	昆明石林风景区	4.6	417.9	大理崇圣寺三塔文化旅游区
0.5	大理崇圣寺三塔文化旅游区	1.9	169.8	丽江古城景区
1	丽江古城景区	1	38.7	丽江玉龙雪山景区
1	丽江玉龙雪山景区	2.5	228.5	迪庆藏族自治州香格里拉普达措国家公园
2	迪庆藏族自治州香格里拉普达措国家公园	12.9	1164	中科院西双版纳热带植物园
2.2	昆明市	17.7	1590	西安市
1.5	西安市	12.1	1090	南昌市
0.5	南昌市	2	182.8	景德镇古窑民俗博览区
1	景德镇古窑民俗博览区	1.1	100.1	上饶婺源县江湾景区
1	上饶婺源县江湾景区	1.3	116.7	上饶三清山旅游景区
1	上饶三清山旅游景区	2.1	184.7	鹰潭市贵溪龙虎山风景名胜景区
1	鹰潭市贵溪龙虎山风景名胜景区	3.4	301.8	赣州市瑞金市共和国摇篮景区
1.5	赣州市瑞金市共和国摇篮景区	3.5	312.5	吉安井冈山风景旅游区
2	吉安井冈山风景旅游区	5	451	九江庐山风景名胜景区
1.5	南昌市	12.1	1090	西安市
0.7	西安市	5.3	480	郑州市
0.5	郑州市	0.8	68.9	开封清明上河园景区
0.5	开封清明上河园景区	1.9	168.1	安阳殷墟景区
1	安阳殷墟景区	2.9	263.9	郑州登封嵩山少林景区
0.5	郑州登封嵩山少林景区	0.6	54.9	洛阳龙门石窟景区
0.5	洛阳龙门石窟景区	1.7	151.4	洛阳嵩县白云山景区
1	洛阳嵩县白云山景区	0.8	75	洛阳栾川县老君山—鸡冠洞旅游区
1	洛阳栾川县老君山—鸡冠洞旅游区	1.3	114.5	平顶山鲁山县尧山—中原大佛景区
1	平顶山鲁山县尧山—中原大佛景区	1.9	167.7	南阳西峡伏牛山老界岭·恐龙遗址园旅游区
1.5	南阳西峡伏牛山老界岭·恐龙遗址园旅游区	3.4	306.2	洛阳新安县龙潭大峡谷景区
2	洛阳新安县龙潭大峡谷景区	2.2	199.8	焦作(云台山—神农山—青天河)风景区
0.7	郑州市	5.3	480	西安市
3.2	西安市	25.6	2300	哈尔滨市
0.5	哈尔滨市	0.3	10.1	哈尔滨太阳岛景区
1	哈尔滨太阳岛景区	4.1	368	黑河五大连池景区
1	黑河五大连池景区	4.4	400.4	伊春市汤旺河林海奇石景区
1.5	伊春市汤旺河林海奇石景区	7.7	691.8	牡丹江宁安市镜泊湖景区
2	牡丹江宁安市镜泊湖景区	17.2	1546.8	大兴安岭地区漠河北极村旅游景区
3.2	哈尔滨市	25.6	2300	西安市

2.3	西安市	18.3	1650	福州市
0.5	福州市	0.1	2.7	福州市三坊七巷景区
0.5	福州市三坊七巷景区	2.9	261	厦门鼓浪屿风景名胜
1	厦门鼓浪屿风景名胜	1	91.5	泉州市清源山风景名胜
1	泉州市清源山风景名胜	2.1	188.1	福建土楼（永定·南靖）旅游景区
2	福建土楼（永定·南靖）旅游景区	5.2	469.5	三明泰宁风景旅游区
1	三明泰宁风景旅游区	1.8	159.2	南平武夷山风景名胜
1	南平武夷山风景名胜	1.8	165	宁德屏南（白水洋·鸳鸯溪）旅游景区
1	宁德屏南（白水洋·鸳鸯溪）旅游景区	2.3	204.6	宁德市福鼎太姥山旅游区
2.3	福州市	18.3	1650	西安市
2.3	西安市	18.3	1650	广州市
0.5	广州市	0.2	9.2	广州白云山景区
0.5	广州白云山景区	0.6	24.4	佛山市德顺区长鹿旅游休博园
0.5	佛山市德顺区长鹿旅游休博园	0.6	23.3	佛山西樵山景区
0.5	佛山西樵山景区	1.7	149	深圳华侨城旅游度假区
0.5	深圳华侨城旅游度假区	0.8	33	深圳观澜湖休闲旅游区
1	深圳观澜湖休闲旅游区	0.9	79	惠州市罗浮山景区
1	惠州市罗浮山景区	1.2	105.8	广州长隆旅游度假区
1	广州长隆旅游度假区	3.1	280.9	韶关仁化丹霞山景区
0.5	韶关仁化丹霞山景区	2.8	253.7	清远连州地下河旅游景区
1.5	清远连州地下河旅游景区	7.4	669.8	梅州市梅县区雁南飞茶田景区
2.3	广州市	18.3	1650	西安市
1.8	西安市	14.8	1330	杭州市
0.5	杭州市	0.1	6	杭州西湖风景区
0.5	杭州西湖风景区	0	0	杭州西溪湿地旅游区
0.5	杭州西溪湿地旅游区	0.8	69.9	绍兴市鲁迅故里—沈园景区
0.5	绍兴市鲁迅故里—沈园景区	1.2	106.4	嘉兴南湖旅游区
0.5	嘉兴南湖旅游区	0.5	47.4	湖州市南浔区南浔古镇景区
0.5	湖州市南浔区南浔古镇景区	0.5	21.2	嘉兴桐乡乌镇古镇旅游区
0.5	嘉兴桐乡乌镇古镇旅游区	2.3	210.5	宁波奉化溪口—滕头旅游景区
0.5	宁波奉化溪口—滕头旅游景区	1.6	146.1	金华东阳横店影视城景区
0.5	金华东阳横店影视城景区	2.7	244.6	衢州市开化根宫佛国文化旅游区
1	衢州市开化根宫佛国文化旅游区	2	184.3	杭州淳安千岛湖风景区
1.5	杭州淳安千岛湖风景区	4.5	400.9	温州乐清市雁荡山风景区
1.5	温州乐清市雁荡山风景区	3.6	326.5	舟山普陀山风景区
1.8	杭州市	14.8	1330	西安市
1	西安市	8.3	750	成都市
0.5	成都市	1.6	143	乐山乐山大佛景区
1	乐山乐山大佛景区	1.1	97.9	乐山峨眉山景区
1	乐山峨眉山景区	2.9	261.3	成都青城山—都江堰旅游景区
1	成都青城山—都江堰旅游景区	0.5	41.8	阿坝藏族羌族自治州汶川特别旅游区（震中映秀—水磨古镇—三江生态旅游区）
1	阿坝藏族羌族自治州汶川特别旅游区（震中映秀—水磨古镇—三江生态旅游区）	2.2	199.8	绵阳北川羌城旅游区（中国羌城—老县城地震遗址—“5·12”特大地震纪念馆—北川羌族民俗博物馆—北川新县

				城—吉娜羌寨)
0.5	绵阳北川羌城旅游区(中国羌城—老县城地震遗址—“5·12”特大地震纪念馆—北川羌族民俗博物馆—北川新县城—吉娜羌寨)	2.9	258.8	南充市阆中古城旅游景区
0.5	南充市阆中古城旅游景区	2.1	192.5	广安市邓小平故里旅游区
1.5	广安市邓小平故里旅游区	3.6	326.6	广元市剑门蜀道剑门关旅游景区
1	广元市剑门蜀道剑门关旅游景区	3.6	323.4	阿坝藏族羌族自治州松潘县黄龙风景名胜区
1	阿坝藏族羌族自治州松潘县黄龙风景名胜区	1.4	127.1	阿坝藏族羌族自治州九寨沟景区
1	成都市	8.3	750	西安市
1.4	西安市	11	990	武汉市
0.5	武汉市	0.2	9.7	武汉黄鹤楼公园
0.5	武汉黄鹤楼公园	0.8	31.4	武汉市东湖景区
0.5	武汉市东湖景区	1.2	103.6	武汉市黄陂木兰文化生态旅游区
1	武汉市黄陂木兰文化生态旅游区	4.6	416.3	宜昌长阳县清江画廊景区
0.5	宜昌长阳县清江画廊景区	0.9	77.2	宜昌秭归县屈原故里文化旅游区
0.5	宜昌秭归县屈原故里文化旅游区	0.4	15.9	宜昌三峡大坝旅游区
0.5	宜昌三峡大坝旅游区	1.6	148	恩施土家族苗族自治州巴东神龙溪纤夫文化旅游区
1	恩施土家族苗族自治州巴东神龙溪纤夫文化旅游区	2.1	190.9	宜昌三峡人家风景区
1.5	宜昌三峡人家风景区	4	357.5	十堰丹江口市武当山风景区
2	十堰丹江口市武当山风景区	2.1	192.7	神农架生态旅游区
1.5	神农架生态旅游区	4.5	404.2	恩施土家族苗族自治州恩施大峡谷景区
1.4	武汉市	11	990	西安市
3.5	西安市	28.2	2540	乌鲁木齐市
0.5	乌鲁木齐市	2.1	189.1	吐鲁番葡萄沟风景区
1	吐鲁番葡萄沟风景区	2.3	207.1	乌鲁木齐天山大峡谷
1	乌鲁木齐天山大峡谷	1.6	145.7	昌吉州阜康市天山天池风景名胜区
1.5	昌吉州阜康市天山天池风景名胜区	4.5	409	巴音郭楞蒙古自治州博湖县博斯腾湖景区
1.5	巴音郭楞蒙古自治州博湖县博斯腾湖景区	4	356.7	伊犁地区新源县那拉提旅游风景区
1.5	伊犁地区新源县那拉提旅游风景区	11.6	1042.1	喀什地区噶尔老城景区
1	喀什地区噶尔老城景区	2.4	218.3	喀什地区泽普县金胡杨景区
2	喀什地区泽普县金胡杨景区	10.3	923.1	阿勒泰地区富蕴县可可托海景区
2	阿勒泰地区富蕴县可可托海景区	5	450.8	阿勒泰地区布尔津县喀纳斯景区
3.5	乌鲁木齐市	28.2	2540	西安市
1.5	西安市	12.1	1090	南京市
0.5	南京市	0.3	12	南京夫子庙—秦淮风光带(江南贡院-白鹭洲-中华门-瞻园-王谢故居)
0.5	南京夫子庙—秦淮风光带	0.2	7.1	南京钟山—中山陵风景名胜区(明孝陵

	(江南贡院-白鹭洲-中华门-瞻园-王谢故居)			-音乐台-灵谷寺-梅花山-紫金山天文台)
0.5	南京钟山—中山陵风景名胜区(明孝陵-音乐台-灵谷寺-梅花山-紫金山天文台)	1.1	96.7	扬州瘦西湖风景区
0.5	扬州瘦西湖风景区	0.9	81.3	泰州姜堰区溱湖国家湿地公园
0.5	泰州姜堰区溱湖国家湿地公园	0.8	33.8	苏州昆山周庄古镇景区
0.5	苏州昆山周庄古镇景区	0.5	44.3	苏州园林(拙政园—留园—虎丘)
0.5	苏州园林(拙政园—留园—虎丘)	0.3	11.1	苏州市金鸡湖国家商务旅游示范区
0.5	苏州市金鸡湖国家商务旅游示范区	0.7	26.3	苏州吴江同里古镇景区
0.5	苏州吴江同里古镇景区	0.9	83.2	中央电视台无锡影视基地三国水浒城景区
0.5	中央电视台无锡影视基地三国水浒城景区	0.1	6	无锡鼋头渚景区
0.5	无锡鼋头渚景区	0.6	56.9	常州环球恐龙城景区(中华恐龙园-恐龙谷温泉-恐龙城大剧院)
0.5	常州环球恐龙城景区(中华恐龙园-恐龙谷温泉-恐龙城大剧院)	0.9	79.9	无锡灵山大佛景区
0.5	无锡灵山大佛景区	1.9	167.8	南通市濠河风景区
0.5	南通市濠河风景区	2.6	231.6	淮安市周恩来故里景区(周恩来纪念馆-周恩来故居-附马巷历史街区-河下古镇)
1	淮安市周恩来故里景区(周恩来纪念馆-周恩来故居-附马巷历史街区-河下古镇)	0.7	63.5	镇江三山风景名胜区(金山—北固山—焦山)
1	镇江三山风景名胜区(金山—北固山—焦山)	1	87.9	镇江句容茅山景区
1	镇江句容茅山景区	1	86.5	常州溧阳市天目湖景区(天目湖-南山竹海-御水温泉)
1	常州溧阳市天目湖景区(天目湖-南山竹海-御水温泉)	1.6	146	苏州常熟沙家浜—虞山尚湖旅游区
1	苏州常熟沙家浜—虞山尚湖旅游区	0.8	71.6	苏州吴中太湖旅游区(旺山—穹窿山—东山)
1.5	南京市	12.1	1090	西安市

表 3 第三问的结果

消耗时间(天)	出发方式	出发地	费用(元)	路途时间(h)	游览景区
0.5	租车自驾	北京市	301	0.3	故宫博物院
0.5	租车自驾	故宫博物院	302.9	0.3	恭王府景区
0.5	租车自驾	恭王府景区	305.1	0.2	天坛公园
0.5	租车自驾	天坛公园	311.9	0.2	北京奥林匹克公园(鸟巢-水立方-中国科技馆-国家奥林匹克森林

					公园)
0.5	租车自驾	北京奥林匹克公园(鸟巢-水立方-中国科技馆-国家奥林匹克森林公园)	307.7	0.3	颐和园
0.5	租车自驾	颐和园	321.5	0.2	明十三陵景区(神路-定陵-长陵-昭陵)
0.5	租车自驾	明十三陵景区(神路-定陵-长陵-昭陵)	348.3	0.1	八达岭—慕田峪长城旅游区
0.6	高铁	北京市	163.5	0.7	天津市
0.5	租车自驾	天津市	152.8	0.1	天津古文化街旅游区(津门故里)
1	租车自驾	天津古文化街旅游区(津门故里)	424.4	0.2	天津蓟县盘山风景名胜区
1	高铁	天津市	0	0	石家庄市
1	租车自驾	石家庄市	382.2	0.3	石家庄平山县西柏坡景区
1	租车自驾	石家庄平山县西柏坡景区	533.4	0.3	保定安新白洋淀景区(文化苑-大观园-鸳鸯岛-元妃荷园-嘎子印象-渔人乐园)
1	租车自驾	保定安新白洋淀景区(文化苑-大观园-鸳鸯岛-元妃荷园-嘎子印象-渔人乐园)	613.3	0.3	保定涞水县野三坡景区(百里峡-白草畔-鱼谷洞-龙门天关)
1.5	租车自驾	保定涞水县野三坡景区(百里峡-白草畔-鱼谷洞-龙门天关)	1004.4	4	承德避暑山庄及周围寺庙景区(普陀宗乘-须弥福寺-普宁寺-普佑寺)
1	租车自驾	承德避暑山庄及周围寺庙景区(普陀宗乘-须弥福寺-普宁寺-普佑寺)	689.7	0.2	秦皇岛山海关景区(老龙头-山海关古城-天下第一关-孟姜女庙)
1.2	高铁	石家庄市	385.5	1.5	北京市
0.2	高铁	北京市	553.5	1.7	济南市
0.5	租车自驾	济南市	361	0.1	济宁曲阜明故城三孔旅游区
0.5	租车自驾	济宁曲阜明故城三孔旅游区	451.3	0.2	枣庄台儿庄古城景区
1	租车自驾	枣庄台儿庄古城景区	667	0.3	泰安泰山景区
1	租车自驾	泰安泰山景区	532.2	0.2	济南天下第一泉景区(趵突泉—大明湖—五龙潭—环城公园-黑虎泉)
1	租车自驾	济南天下第一泉景区(趵突泉—大明湖—五龙潭—环城公园-黑虎泉)	799.7	0.3	山东沂蒙山旅游区(沂山景区—龟蒙景区—云蒙景区)
0.5	租车自驾	山东沂蒙山旅游区(沂山景区—龟蒙景区—云蒙景区)	428.4	0.1	青岛崂山景区

0.5	租车自驾	青岛崂山景区	440	0.1	烟台蓬莱阁—三仙山—八仙过海旅游区
0.5	租车自驾	烟台蓬莱阁—三仙山—八仙过海旅游区	348.7	0.1	烟台龙口南山景区
0.5	租车自驾	烟台龙口南山景区	397.6	0.1	威海刘公岛景区
1.5	高铁	济南市	1195.5	3.9	上海市
0.5	租车自驾	上海市	303.5	0.3	东方明珠广播电视塔
0.5	租车自驾	东方明珠广播电视塔	302.1	0.1	上海科技馆
1	租车自驾	上海科技馆	603.1	0.2	上海野生动物园
1.1	高铁	上海市	1659	5.2	北京市
0.5	高铁	北京市	945	4	郑州市
0.5	租车自驾	郑州市	293.9	0.2	开封清明上河园景区
0.5	租车自驾	开封清明上河园景区	393.1	0.1	安阳殷墟景区
1	租车自驾	安阳殷墟景区	736.3	4	洛阳龙门石窟景区
1	租车自驾	洛阳龙门石窟景区	428.8	0.3	平顶山鲁山县尧山—中原大佛景区
0.5	租车自驾	平顶山鲁山县尧山—中原大佛景区	384.1	0.3	洛阳嵩县白云山景区
1	租车自驾	洛阳嵩县白云山景区	651.7	0.3	郑州登封嵩山少林景区
1	租车自驾	郑州登封嵩山少林景区	732.6	0.3	洛阳新安县龙潭大峡谷景区
1	租车自驾	洛阳新安县龙潭大峡谷景区	816.1	0.1	洛阳栾川县老君山—鸡冠洞旅游区
1	租车自驾	洛阳栾川县老君山—鸡冠洞旅游区	719	0.3	南阳西峡伏牛山老界岭·恐龙遗址园旅游区
2.5	租车自驾	南阳西峡伏牛山老界岭·恐龙遗址园旅游区	1511.4	4	焦作(云台山—神农山—青天河)风景区
1.5	高铁	郑州市	945	4	北京市
0.6	高铁	北京市	1282.5	4.4	合肥市
0.5	租车自驾	合肥市	320.2	0.1	阜阳市颍上县八里河风景区
1.5	租车自驾	阜阳市颍上县八里河风景区	721.4	4	安庆潜山县天柱山风景区
1	租车自驾	安庆潜山县天柱山风景区	476.1	0.1	池州青阳县九华山风景区
0.5	租车自驾	池州青阳县九华山风景区	368.7	0.2	黄山市古徽州文化旅游区(徽州古城—牌坊群鲍家花园—唐模—潜口民宅—呈坎)
1	租车自驾	黄山市古徽州文化旅游区(徽州古城—牌坊群鲍家花园—唐模—潜口民宅—呈坎)	353.4	0.3	黄山市黟县皖南古村落—西递宏村
1	租车自驾	黄山市黟县皖南古村落—西递宏村	402.6	0.2	宣城市绩溪县龙川景区

1	租车自驾	宣城市绩溪县龙川景区	544.8	0.2	黄山市黄山风景区
1.5	租车自驾	黄山市黄山风景区	848.8	4	六安市金寨县天堂寨旅游景区
1.6	高铁	合肥市	1282.5	4.4	北京市
0.7	高铁	北京市	1546.5	5.6	西安市
0.5	租车自驾	西安市	304.6	0.1	西安大雁塔—大唐芙蓉园景区
0.5	租车自驾	西安大雁塔—大唐芙蓉园景区	322.6	0.3	西安华清池景区
0.5	租车自驾	西安华清池景区	305.5	0.1	西安秦始皇兵马俑博物馆
0.5	租车自驾	西安秦始皇兵马俑博物馆	382	0.3	宝鸡扶风县法门寺佛文化景区
1	租车自驾	宝鸡扶风县法门寺佛文化景区	838.1	0.1	渭南华阴市华山风景区
1	租车自驾	渭南华阴市华山风景区	882	4	延安黄陵县黄帝陵景区
0.9	高铁	西安市	535.5	3	太原市
0.5	租车自驾	太原市	234.7	0.1	大同云冈石窟
1	租车自驾	大同云冈石窟	497.3	0.3	忻州五台山风景名胜景区
0.5	租车自驾	忻州五台山风景名胜景区	419.9	0.1	晋中市乔家大院文化园区
0.5	租车自驾	晋中市乔家大院文化园区	171	0.2	晋中市平遥县平遥古城景区
0.5	租车自驾	晋中市平遥县平遥古城景区	282.9	0.3	晋中市介休市绵山风景名胜景区
0.5	租车自驾	晋中市介休市绵山风景名胜景区	415.6	0.2	晋城阳城县皇城相府生态文化旅游区
0.6	飞机	太原市	3210	0.6	北京市
0.6	高铁	北京市	1561.5	5.2	长沙市
0.5	租车自驾	长沙市	193.3	0.3	长沙市宁乡县花明楼景区
0.5	租车自驾	长沙市宁乡县花明楼景区	270.9	0.3	长沙岳麓山—橘子洲旅游区
0.5	租车自驾	长沙岳麓山—橘子洲旅游区	359.9	0.2	衡阳南岳衡山旅游区
0.5	租车自驾	衡阳南岳衡山旅游区	416.2	0.1	郴州市东江湖旅游区
1	租车自驾	郴州市东江湖旅游区	920.1	4	湘潭韶山旅游区
0.5	租车自驾	湘潭韶山旅游区	478.7	0.1	岳阳岳阳楼—君山岛景区
2.5	租车自驾	岳阳岳阳楼—君山岛景区	1452	4	张家界武陵源—天门山旅游区
0.7	飞机	长沙市	3630	1.5	呼和浩特市
1	租车自驾	呼和浩特市	634.2	0.2	鄂尔多斯达拉特旗响沙湾旅游景区
0.5	租车自驾	鄂尔多斯达拉特旗响沙湾旅游景区	337.4	0.3	鄂尔多斯伊金霍洛旗成吉思汗陵旅游区

1.1	飞机	呼和浩特市	1650	0.5	北京市
0.8	高铁	北京市	1878	6.5	杭州市
0.5	租车自驾	杭州市	289.3	0.1	绍兴市鲁迅故里—沈园景区
0.5	租车自驾	绍兴市鲁迅故里—沈园景区	331.4	0.2	嘉兴南湖旅游区
0.5	租车自驾	嘉兴南湖旅游区	247.4	0.2	嘉兴桐乡乌镇古镇旅游区
0.5	租车自驾	嘉兴桐乡乌镇古镇旅游区	237.7	0.2	湖州市南浔区南浔古镇景区
0.5	租车自驾	湖州市南浔区南浔古镇景区	398.1	0.3	杭州西溪湿地旅游区
0.5	租车自驾	杭州西溪湿地旅游区	300	0.3	杭州西湖风景区
0.5	租车自驾	杭州西湖风景区	371.1	0.1	金华东阳横店影视城景区
0.5	租车自驾	金华东阳横店影视城景区	371.1	0.1	宁波奉化溪口—滕头旅游景区
1	租车自驾	宁波奉化溪口—滕头旅游景区	602	0.2	舟山普陀山风景区
1.5	租车自驾	舟山普陀山风景区	1001.5	4	温州乐清市雁荡山风景区
1	租车自驾	温州乐清市雁荡山风景区	845	4	衢州市开化根宫佛国文化旅游区
1	租车自驾	衢州市开化根宫佛国文化旅游区	784.3	0.2	杭州淳安千岛湖风景区
1.3	高铁	杭州市	1878	6.5	北京市
0.6	飞机	北京市	3150	1.1	长春市
1	租车自驾	长春市	306.2	0.3	长白山景区
0.5	租车自驾	长白山景区	302.2	0.2	长春净月潭景区
0.5	租车自驾	长春净月潭景区	303.5	0.2	长春市长影世纪城景区
0.5	租车自驾	长春市长影世纪城景区	313	0.3	长春伪满皇宫博物馆
0.1	高铁	长春市	220.5	1.1	哈尔滨市
0.5	租车自驾	哈尔滨市	306.1	0.1	哈尔滨太阳岛景区
1	租车自驾	哈尔滨太阳岛景区	818	4	黑河五大连池景区
1	租车自驾	黑河五大连池景区	1000.4	4	伊春市汤旺河林海奇石景区
1.5	租车自驾	伊春市汤旺河林海奇石景区	1366.8	8	牡丹江宁安市镜泊湖景区
2	租车自驾	牡丹江宁安市镜泊湖景区	2877.2	8	大兴安岭地区漠河北极村旅游景区
0.7	飞机	哈尔滨市	3270	1.4	北京市
0.2	飞机	北京市	3540	1.3	银川市
0.5	租车自驾	银川市	315	0.3	银川市灵武水洞沟旅游区
0.5	租车自驾	银川市灵武水洞沟旅游区	356.9	0.1	银川镇北堡西部影视城
1	租车自驾	银川镇北堡西部影视城	354.4	0.2	石嘴山平罗县沙湖旅游景区

0.5	租车自驾	石嘴山平罗县沙湖旅游景区	564.4	0.3	中卫沙坡头旅游景区
0.1	飞机	银川市	1500	0.7	西宁市
0.5	租车自驾	西宁市	165.3	0.1	西宁市湟中县塔尔寺景区
1	租车自驾	西宁市湟中县塔尔寺景区	616.5	0.1	青海湖风景区
1.2	飞机	西宁市	5220	1.9	北京市
0.2	飞机	北京市	4170	1.7	兰州市
1	租车自驾	兰州市	776.6	4	天水麦积山景区
1	租车自驾	天水麦积山景区	673.4	0.1	平凉崆峒山风景名胜区
1.5	租车自驾	平凉崆峒山风景名胜区	1726.9	8	嘉峪关文物景区
1.5	租车自驾	嘉峪关文物景区	1042.1	4	酒泉市敦煌沙山月牙泉景区
0.7	飞机	兰州市	4170	1.7	北京市
0.2	飞机	北京市	4290	1.6	南昌市
0.5	租车自驾	南昌市	407.8	0.3	景德镇古窑民俗博览区
1	租车自驾	景德镇古窑民俗博览区	400.1	0.1	上饶婺源县江湾景区
1	租车自驾	上饶婺源县江湾景区	416.7	0.3	上饶三清山旅游景区
1	租车自驾	上饶三清山旅游景区	634.7	0.1	鹰潭市贵溪龙虎山风景名胜区
1	租车自驾	鹰潭市贵溪龙虎山风景名胜区	751.8	4	赣州市瑞金市共和国摇篮景区
1.5	租车自驾	赣州市瑞金市共和国摇篮景区	987.5	4	吉安井冈山风景旅游区
2	租车自驾	吉安井冈山风景旅游区	1351	8	九江庐山风景名胜区
1.2	飞机	南昌市	4290	1.6	北京市
0.6	飞机	北京市	4410	0.8	沈阳市
0.5	租车自驾	沈阳市	303.4	0.2	沈阳植物园
1	租车自驾	沈阳植物园	401.9	0.1	本溪市本溪水洞景区
1	租车自驾	本溪市本溪水洞景区	868.9	4	大连金石滩景区(地质公园-发现王国-蜡像馆-文化博览广场)
0.5	租车自驾	大连金石滩景区(地质公园-发现王国-蜡像馆-文化博览广场)	282.6	0.3	大连老虎滩海洋公园—老虎滩极地馆
0.6	飞机	沈阳市	4410	0.8	北京市
0.2	飞机	北京市	4920	1.9	重庆市
0.5	租车自驾	重庆市	396.1	0.2	大足石刻景区
1	租车自驾	大足石刻景区	800.3	0.3	南川金佛山—神龙峡风景区
1	租车自驾	南川金佛山—神龙峡风景区	652.1	0.3	万盛黑山谷-龙鳞石海风景区
1.5	租车自驾	万盛黑山谷-龙鳞石海风景区	1226.5	4	酉阳桃花源旅游景区

1.5	租车自驾	酉阳桃花源旅游景区	1424.2	8	巫山小三峡—小小三峡旅游区
3	租车自驾	巫山小三峡—小小三峡旅游区	2387.6	8	武隆喀斯特旅游区（天生三硚、仙女山、芙蓉洞）
0.7	飞机	重庆市	4920	1.9	北京市
0.3	飞机	北京市	5040	2.1	福州市
0.5	租车自驾	福州市	226.6	0.2	福州市三坊七巷景区
0.5	租车自驾	福州市三坊七巷景区	486.8	0.1	厦门鼓浪屿风景名胜景区
1	租车自驾	厦门鼓浪屿风景名胜景区	541.5	0.1	泉州市清源山风景名胜景区
1	租车自驾	泉州市清源山风景名胜景区	788.1	0.3	福建土楼（永定·南靖）旅游景区
2	租车自驾	福建土楼（永定·南靖）旅游景区	1452.6	8	南平武夷山风景名胜景区
1	租车自驾	南平武夷山风景名胜景区	759.2	0.2	三明泰宁风景旅游区
1.5	租车自驾	三明泰宁风景旅游区	1183.6	4	宁德屏南（白水洋·鸳鸯溪）旅游景区
1	租车自驾	宁德屏南（白水洋·鸳鸯溪）旅游景区	804.6	0.3	宁德市福鼎太姥山旅游区
0.8	飞机	福州市	5040	2.1	北京市
0.3	飞机	北京市	5730	2.4	广州市
0.5	租车自驾	广州市	159.7	0.2	佛山市德顺区长鹿旅游休博园
0.5	租车自驾	佛山市德顺区长鹿旅游休博园	239	0.3	佛山西樵山景区
0.5	租车自驾	佛山西樵山景区	280.8	0.1	广州白云山景区
0.5	租车自驾	广州白云山景区	349.1	0.1	深圳观澜湖休闲旅游区
0.5	租车自驾	深圳观澜湖休闲旅游区	244.8	0.3	深圳华侨城旅游度假区
1	租车自驾	深圳华侨城旅游度假区	420.4	0.1	惠州市罗浮山景区
1	租车自驾	惠州市罗浮山景区	555.8	0.3	广州长隆旅游度假区
1	租车自驾	广州长隆旅游度假区	580.9	4	韶关仁化丹霞山景区
0.5	租车自驾	韶关仁化丹霞山景区	478.7	0.2	清远连州地下河旅游景区
1.5	租车自驾	清远连州地下河旅游景区	1344.8	8	梅州市梅县区雁南飞茶田景区
1.3	飞机	广州市	5730	2.4	北京市
0.3	飞机	北京市	5940	2.4	贵阳市
0.5	租车自驾	贵阳市	338.5	0.1	安顺龙宫景区
0.5	租车自驾	安顺龙宫景区	273.9	0.1	安顺镇宁县黄果树瀑布景区
1	租车自驾	安顺镇宁县黄果树瀑布景区	841.1	0.1	毕节市百里杜鹃景区
1.5	租车自驾	毕节市百里杜鹃景区	1352.6	4	黔南布依族苗族自治州荔波樟江景区

0.8	飞机	贵阳市	5940	2.4	北京市
0.3	飞机	北京市	6450	2.6	南宁市
0.5	租车自驾	南宁市	302.9	0.1	南宁市青秀山旅游区
1	租车自驾	南宁市青秀山旅游区	838.8	4	桂林独秀峰·靖江王城景区
1	租车自驾	桂林独秀峰·靖江王城景区	614.7	0.3	桂林漓江风景区
1	租车自驾	桂林漓江风景区	694.4	0.3	桂林兴安县乐满地度假世界
1.3	飞机	南宁市	6450	2.6	北京市
0.4	飞机	北京市	7050	3	海口市
0.5	租车自驾	海口市	480.6	0.1	保亭县呀诺达雨林文化旅游区
0.5	租车自驾	保亭县呀诺达雨林文化旅游区	230.2	0.3	保亭县海南槟榔谷黎苗文化旅游区
0.5	租车自驾	保亭县海南槟榔谷黎苗文化旅游区	287.1	0.3	三亚南山文化旅游区
0.5	租车自驾	三亚南山文化旅游区	230.6	0.3	三亚南山大小洞天旅游区
0.5	租车自驾	三亚南山大小洞天旅游区	425.1	0.2	陵水县分界洲岛旅游区
0.9	飞机	海口市	7050	3	北京市
0.5	飞机	北京市	8790	4.1	拉萨市
0.5	租车自驾	拉萨市	302.1	0.1	拉萨大昭寺景区
0.5	租车自驾	拉萨大昭寺景区	300.9	0.1	拉萨布达拉宫景区
1	飞机	拉萨市	8790	4.1	北京市
0.5	高铁	北京市	1330.5	3.9	南京市
0.5	租车自驾	南京市	307.2	0.3	南京夫子庙—秦淮河风光带(江南贡院-白鹭洲-中华门-瞻园-王谢故居)
0.5	租车自驾	南京夫子庙—秦淮河风光带(江南贡院-白鹭洲-中华门-瞻园-王谢故居)	304.3	0.1	南京钟山—中山陵风景名胜区(明孝陵-音乐台-灵谷寺-梅花山-紫金山天文台)
0.5	租车自驾	南京钟山—中山陵风景名胜区(明孝陵-音乐台-灵谷寺-梅花山-紫金山天文台)	321.7	0.1	扬州瘦西湖风景区
0.5	租车自驾	扬州瘦西湖风景区	306.3	0.1	泰州姜堰区溱湖国家湿地公园
0.5	租车自驾	泰州姜堰区溱湖国家湿地公园	245.3	0.2	苏州昆山周庄古镇景区
0.5	租车自驾	苏州昆山周庄古镇景区	269.3	0.1	苏州园林(拙政园—留园—虎丘)
0.5	租车自驾	苏州园林(拙政园—留园—虎丘)	231.7	0.3	苏州市金鸡湖国家商务旅游示范区
0.5	租车自驾	苏州市金鸡湖国家商务旅游示范区	240.8	0.1	苏州吴江同里古镇景区

0.5	租车自驾	苏州吴江同里古镇景区	308.3	0.3	无锡鼋头渚景区
0.5	租车自驾	无锡鼋头渚景区	244.1	0.3	无锡灵山大佛景区
0.5	租车自驾	无锡灵山大佛景区	304.9	0.1	常州环球恐龙城景区(中华恐龙园-恐龙谷温泉-恐龙城大剧院)
0.5	租车自驾	常州环球恐龙城景区(中华恐龙园-恐龙谷温泉-恐龙城大剧院)	354.6	0.2	中央电视台无锡影视基地三国水浒城景区
0.5	租车自驾	中央电视台无锡影视基地三国水浒城景区	325.4	0.3	南通市濠河风景区
0.5	租车自驾	南通市濠河风景区	495.5	0.2	淮安市周恩来故里景区(周恩来纪念馆-周恩来故居-附马巷历史街区-河下古镇)
1	租车自驾	淮安市周恩来故里景区(周恩来纪念馆-周恩来故居-附马巷历史街区-河下古镇)	513.5	0.3	镇江三山风景名胜区(金山—北固山—焦山)
1	租车自驾	镇江三山风景名胜区(金山—北固山—焦山)	537.9	0.2	镇江句容茅山景区
1	租车自驾	镇江句容茅山景区	536.5	0.1	常州溧阳市天目湖景区(天目湖-南山竹海-御水温泉)
1	租车自驾	常州溧阳市天目湖景区(天目湖-南山竹海-御水温泉)	596	0.2	苏州常熟沙家浜—虞山尚湖旅游区
1	租车自驾	苏州常熟沙家浜—虞山尚湖旅游区	521.6	0.2	苏州吴中太湖旅游区(旺山—穹窿山—东山)
0.2	飞机	北京市	4350	1.6	武汉市
0.5	租车自驾	武汉市	230.8	0.2	武汉黄鹤楼公园
0.5	租车自驾	武汉黄鹤楼公园	243.8	0.1	武汉市东湖景区
0.5	租车自驾	武汉市东湖景区	403.6	0.3	武汉市黄陂木兰文化生态旅游区
1	租车自驾	武汉市黄陂木兰文化生态旅游区	731	4	宜昌秭归县屈原故里文化旅游区
0.5	租车自驾	宜昌秭归县屈原故里文化旅游区	234.5	0.2	宜昌三峡大坝旅游区
0.5	租车自驾	宜昌三峡大坝旅游区	398.1	0.1	宜昌长阳县清江画廊景区
0.5	租车自驾	宜昌长阳县清江画廊景区	534.2	0.2	恩施土家族苗族自治州巴东神龙溪纤夫文化旅游区
1	租车自驾	恩施土家族苗族自治州巴东神龙溪纤夫文化旅游区	640.9	0.3	宜昌三峡人家风景区
1.5	租车自驾	宜昌三峡人家风景区	1104.8	4	十堰丹江口市武当山风景区
2	租车自驾	十堰丹江口市武当山	1392.7	0.1	神农架生态旅游区

		风景区			
1.5	租车自驾	神农架生态旅游区	1304.2	4	恩施土家族苗族自治州恩施大峡谷景区
0.3	飞机	北京市	5070	2	成都市
0.5	租车自驾	成都市	368	0.1	乐山乐山大佛景区
1	租车自驾	乐山乐山大佛景区	651.1	0.3	成都青城山—都江堰旅游景区
1	租车自驾	成都青城山—都江堰旅游景区	608	0.3	绵阳北川羌城旅游区(中国羌城—老县城地震遗址—“5·12”特大地震纪念馆—北川羌族民俗博物馆—北川新县城—吉娜羌寨)
1	租车自驾	绵阳北川羌城旅游区(中国羌城—老县城地震遗址—“5·12”特大地震纪念馆—北川羌族民俗博物馆—北川新县城—吉娜羌寨)	459.4	0.2	广元市剑门蜀道剑门关旅游景区
0.5	租车自驾	广元市剑门蜀道剑门关旅游景区	377.6	0.3	南充市阆中古城旅游景区
0.5	租车自驾	南充市阆中古城旅游景区	417.5	0.3	广安市邓小平故里旅游区
1.5	租车自驾	广安市邓小平故里旅游区	1302.1	4	阿坝藏族羌族自治州汶川特别旅游区(震中映秀—水磨古镇—三江生态旅游区)
1	租车自驾	阿坝藏族羌族自治州汶川特别旅游区(震中映秀—水磨古镇—三江生态旅游区)	913.6	4	阿坝藏族羌族自治州松潘县黄龙风景名胜区
1	租车自驾	阿坝藏族羌族自治州松潘县黄龙风景名胜区	727.1	0.1	阿坝藏族羌族自治州九寨沟景区
2	租车自驾	阿坝藏族羌族自治州九寨沟景区	1818.4	8	乐山峨眉山景区
0.4	飞机	北京市	6540	2.9	昆明市
0.5	租车自驾	昆明市	309	0.3	昆明石林风景区
1	租车自驾	昆明石林风景区	868.9	4	大理崇圣寺三塔文化旅游区
0.5	租车自驾	大理崇圣寺三塔文化旅游区	394.8	0.2	丽江古城景区
1	租车自驾	丽江古城景区	473.2	0.3	丽江玉龙雪山景区
1	租车自驾	丽江玉龙雪山景区	528.5	0.2	迪庆藏族自治州香格里拉普达措国家公园
2	租车自驾	迪庆藏族自治州香格里拉普达措国家公园	1764	8	中科院西双版纳热带植物园
0.4	飞机	北京市	7890	3.5	乌鲁木齐市
0.5	租车自驾	乌鲁木齐市	489.1	0.1	吐鲁番葡萄沟风景区
1	租车自驾	吐鲁番葡萄沟风景区	657.1	0.3	乌鲁木齐天山大峡谷

1	租车自驾	乌鲁木齐天山大峡谷	595.7	0.1	昌吉州阜康市天山天池风景名胜 区
1.5	租车自驾	昌吉州阜康市天山天 池风景名胜区	1431.4	4	巴音郭楞蒙古自治州博湖县博 斯腾湖景区
1.5	租车自驾	巴音郭楞蒙古自治州 博湖县波斯腾湖景区	1256.7	4	伊犁地区新源县那拉提旅游风 景区
2	租车自驾	伊犁地区新源县那拉 提旅游风景区	1672.2	8	喀什地区泽普县金胡杨景区
0.5	租车自驾	喀什地区泽普县金胡 杨景区	443.3	0.3	喀什地区噶尔老城景区
2	租车自驾	喀什地区噶尔老城景 区	3042.5	8	阿勒泰地区布尔津县喀纳斯景 区
2	租车自驾	阿勒泰地区布尔津县 喀纳斯景区	1650.8	8	阿勒泰地区富蕴县可可托海景 区

表 4 问题 4 的结果

消耗 时间 (天)	出发方式	出发地	费用(元)	路途 时间 (h)	游览景区
0.5	租车自驾	南京市	307.2	0.3	南京夫子庙—秦淮风光带(江南 贡院-白鹭洲-中华门-瞻园-王谢故 居)
0.5	租车自驾	南京夫子庙—秦淮风光 带(江南贡院-白鹭洲-中华 门-瞻园-王谢故居)	304.3	0.3	南京钟山—中山陵风景名胜(明 孝陵-音乐台-灵谷寺-梅花山-紫金 山天文台)
0.5	租车自驾	南京钟山—中山陵风景名 胜区(明孝陵-音乐台-灵谷 寺-梅花山-紫金山天文台)	321.7	0.2	扬州瘦西湖风景区
0.5	租车自驾	扬州瘦西湖风景区	306.3	0.2	泰州姜堰区溱湖国家湿地公园
0.5	租车自驾	泰州姜堰区溱湖国家湿地 公园	245.3	0.3	苏州昆山周庄古镇景区
0.5	租车自驾	苏州昆山周庄古镇景区	269.3	0.2	苏州园林（拙政园—留园—虎丘）
0.5	租车自驾	苏州园林（拙政园—留园 —虎丘）	231.7	0.1	苏州市金鸡湖国家商务旅游示范区
0.5	租车自驾	苏州市金鸡湖国家商务旅 游示范区	240.8	0.1	苏州吴江同里古镇景区
0.5	租车自驾	苏州吴江同里古镇景区	308.3	0.2	无锡鼋头渚景区
0.5	租车自驾	无锡鼋头渚景区	244.1	0.3	无锡灵山大佛景区
0.5	租车自驾	无锡灵山大佛景区	304.9	0.3	常州环球恐龙城景区(中华恐龙园- 恐龙谷温泉-恐龙城大剧院)
0.5	租车自驾	常州环球恐龙城景区(中 华恐龙园-恐龙谷温泉-恐 龙城大剧院)	354.6	0.3	中央电视台无锡影视基地三国水浒 城景区

0.5	租车自驾	中央电视台无锡影视基地 三国水浒城景区	325.4	0.2	南通市濠河风景区
0.5	租车自驾	南通市濠河风景区	495.5	0.1	淮安市周恩来故里景区（周恩来纪念馆-周恩来故居-附马巷历史街区-河下古镇）
1	租车自驾	淮安市周恩来故里景区 （周恩来纪念馆-周恩来故居-附马巷历史街区-河下古镇）	513.5	0.2	镇江三山风景名胜区（金山—北固山—焦山）
1	租车自驾	镇江三山风景名胜区（金山—北固山—焦山）	537.9	0.3	镇江句容茅山景区
1	租车自驾	镇江句容茅山景区	536.5	0.2	常州溧阳市天目湖景区(天目湖-南山竹海-御水温泉)
1	租车自驾	常州溧阳市天目湖景区 (天目湖-南山竹海-御水温泉)	596	0.3	苏州常熟沙家浜—虞山尚湖旅游区
1	租车自驾	苏州常熟沙家浜—虞山尚湖旅游区	521.6	0.1	苏州吴中太湖旅游区（旺山—穹窿山—东山）
0.6	高铁	南京市	181.5	1	合肥市
0.5	租车自驾	合肥市	320.2	0.1	阜阳市颍上县八里河风景区
1.5	租车自驾	阜阳市颍上县八里河风景区	721.4	4	安庆潜山县天柱山风景区
1	租车自驾	安庆潜山县天柱山风景区	476.1	0.1	池州青阳县九华山风景区
0.5	租车自驾	池州青阳县九华山风景区	368.7	0.1	黄山市古徽州文化旅游区（徽州古城—牌坊群鲍家花园—唐模—潜口民宅—呈坎）
1	租车自驾	黄山市古徽州文化旅游区 （徽州古城—牌坊群鲍家花园—唐模—潜口民宅—呈坎）	353.4	0.3	黄山市黟县皖南古村落—西递宏村
1	租车自驾	黄山市黟县皖南古村落—西递宏村	402.6	0.1	宣城市绩溪县龙川景区
1	租车自驾	宣城市绩溪县龙川景区	544.8	0.2	黄山市黄山风景区
1.5	租车自驾	黄山市黄山风景区	848.8	4	六安市金寨县天堂寨旅游景区
1.3	高铁	合肥市	610.5	2.1	上海市
0.5	租车自驾	上海市	303.5	0.2	东方明珠广播电视塔
0.5	租车自驾	东方明珠广播电视塔	302.1	0.1	上海科技馆
1	租车自驾	上海科技馆	603.1	0.3	上海野生动物园
0.2	高铁	上海市	403.5	1.5	南京市
0.7	高铁	南京市	352.5	1.5	杭州市
0.5	租车自驾	杭州市	289.3	0.3	绍兴市鲁迅故里—沈园景区
0.5	租车自驾	绍兴市鲁迅故里—沈园景区	331.4	0.3	嘉兴南湖旅游区
0.5	租车自驾	嘉兴南湖旅游区	247.4	0.3	嘉兴桐乡乌镇古镇旅游区

0.5	租车自驾	嘉兴桐乡乌镇古镇旅游区	237.7	0.1	湖州市南浔区南浔古镇景区
0.5	租车自驾	湖州市南浔区南浔古镇景区	398.1	0.3	杭州西溪湿地旅游区
0.5	租车自驾	杭州西溪湿地旅游区	300	0.1	杭州西湖风景区
0.5	租车自驾	杭州西湖风景区	371.1	0.1	金华东阳横店影视城景区
0.5	租车自驾	金华东阳横店影视城景区	371.1	0.2	宁波奉化溪口—滕头旅游景区
1	租车自驾	宁波奉化溪口—滕头旅游景区	602	0.3	舟山普陀山风景区
1.5	租车自驾	舟山普陀山风景区	1001.5	4	温州乐清市雁荡山风景区
1	租车自驾	温州乐清市雁荡山风景区	845	4	衢州市开化根宫佛国文化旅游区
1	租车自驾	衢州市开化根宫佛国文化旅游区	784.3	0.2	杭州淳安千岛湖风景区
0.7	高铁	杭州市	352.5	1.5	南京市
1.4	高铁	南京市	498	3.5	长沙市
0.5	租车自驾	长沙市	193.3	0.2	长沙市宁乡县花明楼景区
0.5	租车自驾	长沙市宁乡县花明楼景区	270.9	0.1	长沙岳麓山—橘子洲旅游区
0.5	租车自驾	长沙岳麓山—橘子洲旅游区	359.9	0.3	衡阳南岳衡山旅游区
0.5	租车自驾	衡阳南岳衡山旅游区	416.2	0.1	郴州市东江湖旅游区
1	租车自驾	郴州市东江湖旅游区	920.1	4	湘潭韶山旅游区
0.5	租车自驾	湘潭韶山旅游区	478.7	0.3	岳阳岳阳楼—君山岛景区
2.5	租车自驾	岳阳岳阳楼—君山岛景区	1452	4	张家界武陵源—天门山旅游区
0.7	飞机	长沙市	3420	1.3	天津市
0.5	租车自驾	天津市	152.8	0.1	天津古文化街旅游区（津门故里）
1	租车自驾	天津古文化街旅游区（津门故里）	424.4	0.1	天津蓟县盘山风景名胜区
0.1	飞机	天津市	2550	1	南京市
1.7	高铁	南京市	645	5.3	郑州市
0.5	租车自驾	郑州市	293.9	0.3	开封清明上河园景区
0.5	租车自驾	开封清明上河园景区	393.1	0.1	安阳殷墟景区
1	租车自驾	安阳殷墟景区	736.3	4	洛阳龙门石窟景区
1	租车自驾	洛阳龙门石窟景区	428.8	0.2	平顶山鲁山县尧山—中原大佛景区
0.5	租车自驾	平顶山鲁山县尧山—中原大佛景区	384.1	0.3	洛阳嵩县白云山景区
1	租车自驾	洛阳嵩县白云山景区	651.7	0.2	郑州登封嵩山少林景区
1	租车自驾	郑州登封嵩山少林景区	732.6	0.3	洛阳新安县龙潭大峡谷景区
1	租车自驾	洛阳新安县龙潭大峡谷景区	816.1	0.3	洛阳栾川县老君山—鸡冠洞旅游区
1	租车自驾	洛阳栾川县老君山—鸡冠洞旅游区	719	0.2	南阳西峡伏牛山老界岭·恐龙遗址园旅游区
2.5	租车自驾	南阳西峡伏牛山老界岭·恐龙遗址园旅游区	1511.4	4	焦作(云台山—神农山—青天河)风景区
1.7	高铁	郑州市	645	5.3	南京市

1.3	高铁	南京市	837	2.8	济南市
0.5	租车自驾	济南市	361	0.1	济宁曲阜明故城三孔旅游区
0.5	租车自驾	济宁曲阜明故城三孔旅游区	451.3	0.1	枣庄台儿庄古城景区
1	租车自驾	枣庄台儿庄古城景区	667	0.2	泰安泰山景区
1	租车自驾	泰安泰山景区	532.2	0.3	济南天下第一泉景区（趵突泉—大明湖—五龙潭—环城公园-黑虎泉）
1	租车自驾	济南天下第一泉景区（趵突泉—大明湖—五龙潭—环城公园-黑虎泉）	799.7	0.1	山东沂蒙山旅游区（沂山景区—龟蒙景区—云蒙景区）
0.5	租车自驾	山东沂蒙山旅游区（沂山景区—龟蒙景区—云蒙景区）	428.4	0.2	青岛崂山景区
0.5	租车自驾	青岛崂山景区	440	0.2	烟台蓬莱阁—三仙山—八仙过海旅游区
0.5	租车自驾	烟台蓬莱阁—三仙山—八仙过海旅游区	348.7	0.2	烟台龙口南山景区
0.5	租车自驾	烟台龙口南山景区	397.6	0.3	威海刘公岛景区
1.3	高铁	济南市	837	2.8	南京市
1.6	高铁	南京市	994.5	4.9	武汉市
0.5	租车自驾	武汉市	230.8	0.3	武汉黄鹤楼公园
0.5	租车自驾	武汉黄鹤楼公园	243.8	0.1	武汉市东湖景区
0.5	租车自驾	武汉市东湖景区	403.6	0.1	武汉市黄陂木兰文化生态旅游区
1	租车自驾	武汉市黄陂木兰文化生态旅游区	731	4	宜昌秭归县屈原故里文化旅游区
0.5	租车自驾	宜昌秭归县屈原故里文化旅游区	234.5	0.2	宜昌三峡大坝旅游区
0.5	租车自驾	宜昌三峡大坝旅游区	398.1	0.2	宜昌长阳县清江画廊景区
0.5	租车自驾	宜昌长阳县清江画廊景区	534.2	0.3	恩施土家族苗族自治州巴东神龙溪纤夫文化旅游区
1	租车自驾	恩施土家族苗族自治州巴东神龙溪纤夫文化旅游区	640.9	0.2	宜昌三峡人家风景区
1.5	租车自驾	宜昌三峡人家风景区	1104.8	4	十堰丹江口市武当山风景区
2	租车自驾	十堰丹江口市武当山风景区	1392.7	0.2	神农架生态旅游区
1.5	租车自驾	神农架生态旅游区	1304.2	4	恩施土家族苗族自治州恩施大峡谷景区
1.1	高铁	武汉市	994.5	4.9	南京市
1	高铁	南京市	1107	4.4	南昌市
0.5	租车自驾	南昌市	407.8	0.3	景德镇古窑民俗博览区
1	租车自驾	景德镇古窑民俗博览区	400.1	0.1	上饶婺源县江湾景区
1	租车自驾	上饶婺源县江湾景区	416.7	0.3	上饶三清山旅游景区
1	租车自驾	上饶三清山旅游景区	634.7	0.1	鹰潭市贵溪龙虎山风景名胜區

1	租车自驾	鹰潭市贵溪龙虎山风景名胜 区	751.8	4	赣州市瑞金市共和国摇篮景区
1.5	租车自驾	赣州市瑞金市共和国摇篮 景区	987.5	4	吉安井冈山风景旅游区
2	租车自驾	吉安井冈山风景旅游区	1351	8	九江庐山风景名胜区
0.5	高铁	南昌市	1107	4.4	南京市
1.7	高铁	南京市	1230	5.5	福州市
0.5	租车自驾	福州市	226.6	0.2	福州市三坊七巷景区
0.5	租车自驾	福州市三坊七巷景区	486.8	0.3	厦门鼓浪屿风景名胜区
1	租车自驾	厦门鼓浪屿风景名胜区	541.5	0.1	泉州市清源山风景名胜区
1	租车自驾	泉州市清源山风景名胜区	788.1	0.1	福建土楼（永定·南靖）旅游景区
2	租车自驾	福建土楼（永定·南靖） 旅游景区	1452.6	8	南平武夷山风景名胜区
1	租车自驾	南平武夷山风景名胜区	759.2	0.1	三明泰宁风景旅游区
1.5	租车自驾	三明泰宁风景旅游区	1183.6	4	宁德屏南（白水洋·鸳鸯溪）旅游 景区
1	租车自驾	宁德屏南（白水洋·鸳鸯 溪）旅游景区	804.6	0.3	宁德市福鼎太姥山旅游区
0.7	高铁	福州市	1230	5.5	南京市
1	高铁	南京市	1330.5	3.9	北京市
0.5	租车自驾	北京市	301	0.1	故宫博物院
0.5	租车自驾	故宫博物院	302.9	0.3	恭王府景区
0.5	租车自驾	恭王府景区	305.1	0.1	天坛公园
0.5	租车自驾	天坛公园	311.9	0.2	北京奥林匹克公园(鸟巢-水立方-中 国科技馆-国家奥林匹克森林公园)
0.5	租车自驾	北京奥林匹克公园(鸟巢- 水立方-中国科技馆-国家 奥林匹克森林公园)	307.7	0.1	颐和园
0.5	租车自驾	颐和园	321.5	0.3	明十三陵景区(神路-定陵-长陵-昭 陵)
0.5	租车自驾	明十三陵景区(神路-定陵- 长陵-昭陵)	348.3	0.2	八达岭—慕田峪长城旅游区
0.7	高铁	北京市	1546.5	5.6	西安市
0.5	租车自驾	西安市	304.6	0.3	西安大雁塔—大唐芙蓉园景区
0.5	租车自驾	西安大雁塔—大唐芙蓉园 景区	322.6	0.3	西安华清池景区
0.5	租车自驾	西安华清池景区	305.5	0.2	西安秦始皇兵马俑博物馆
0.5	租车自驾	西安秦始皇兵马俑博物馆	382	0.1	宝鸡扶风县法门寺佛文化景区
1	租车自驾	宝鸡扶风县法门寺佛文化 景区	838.1	0.1	渭南华阴市华山风景区
1	租车自驾	渭南华阴市华山风景区	882	4	延安黄陵县黄帝陵景区
0.2	飞机	西安市	3540	1.2	南京市
0.7	飞机	南京市	2640	1.2	太原市

0.5	租车自驾	太原市	234.7	0.3	大同云冈石窟
1	租车自驾	大同云冈石窟	497.3	0.2	忻州五台山风景名胜区
0.5	租车自驾	忻州五台山风景名胜区	419.9	0.3	晋中市乔家大院文化园区
0.5	租车自驾	晋中市乔家大院文化园区	171	0.2	晋中市平遥县平遥古城景区
0.5	租车自驾	晋中市平遥县平遥古城景区	282.9	0.3	晋中市介休市绵山风景名胜区
0.5	租车自驾	晋中市介休市绵山风景名胜区	415.6	0.1	晋城阳城县皇城相府生态文化旅游区
0.6	飞机	太原市	1410	0.5	呼和浩特市
1	租车自驾	呼和浩特市	634.2	0.1	鄂尔多斯达拉特旗响沙湾旅游景区
0.5	租车自驾	鄂尔多斯达拉特旗响沙湾旅游景区	337.4	0.3	鄂尔多斯伊金霍洛旗成吉思汗陵旅游区
0.2	飞机	呼和浩特市	3750	1.7	南京市
0.6	飞机	南京市	2850	1	石家庄市
1	租车自驾	石家庄市	382.2	0.1	石家庄平山县西柏坡景区
1	租车自驾	石家庄平山县西柏坡景区	533.4	0.3	保定安新白洋淀景区(文化苑-大观园-鸳鸯岛-元妃荷园-嘎子印象-渔人乐园)
1	租车自驾	保定安新白洋淀景区(文化苑-大观园-鸳鸯岛-元妃荷园-嘎子印象-渔人乐园)	613.3	0.2	保定涞水县野三坡景区(百里峡-白草畔-鱼谷洞-龙门天关)
1.5	租车自驾	保定涞水县野三坡景区(百里峡-白草畔-鱼谷洞-龙门天关)	1004.4	4	承德避暑山庄及周围寺庙景区(普陀宗乘-须弥福寺-普宁寺-普佑寺)
1	租车自驾	承德避暑山庄及周围寺庙景区(普陀宗乘-须弥福寺-普宁寺-普佑寺)	689.7	0.1	秦皇岛山海关景区(老龙头-山海关古城-天下第一关-孟姜女庙)
0.1	飞机	石家庄市	2850	1	南京市
0.7	飞机	南京市	3870	1.5	广州市
0.5	租车自驾	广州市	159.7	0.1	佛山市德顺区长鹿旅游休博园
0.5	租车自驾	佛山市德顺区长鹿旅游休博园	239	0.1	佛山西樵山景区
0.5	租车自驾	佛山西樵山景区	280.8	0.1	广州白云山景区
0.5	租车自驾	广州白云山景区	349.1	0.3	深圳观澜湖休闲旅游区
0.5	租车自驾	深圳观澜湖休闲旅游区	244.8	0.3	深圳华侨城旅游度假区
1	租车自驾	深圳华侨城旅游度假区	420.4	0.1	惠州市罗浮山景区
1	租车自驾	惠州市罗浮山景区	555.8	0.3	广州长隆旅游度假区
1	租车自驾	广州长隆旅游度假区	580.9	4	韶关仁化丹霞山景区
0.5	租车自驾	韶关仁化丹霞山景区	478.7	0.3	清远连州地下河旅游景区
1.5	租车自驾	清远连州地下河旅游景区	1344.8	8	梅州市梅县区雁南飞茶田景区
0.7	飞机	广州市	3870	1.5	南京市
0.2	飞机	南京市	4050	1.6	重庆市
0.5	租车自驾	重庆市	396.1	0.3	大足石刻景区

1	租车自驾	大足石刻景区	800.3	0.2	南川金佛山—神龙峡风景区
1	租车自驾	南川金佛山—神龙峡风景区	652.1	0.1	万盛黑山谷-龙鳞石海风景区
1.5	租车自驾	万盛黑山谷-龙鳞石海风景区	1226.5	4	酉阳桃花源旅游景区
1.5	租车自驾	酉阳桃花源旅游景区	1424.2	8	巫山小三峡—小小三峡旅游区
3	租车自驾	巫山小三峡—小小三峡旅游区	2387.6	8	武隆喀斯特旅游区（天生三桥、仙女山、芙蓉洞）
0.2	飞机	重庆市	4050	1.6	南京市
0.2	飞机	南京市	4140	1.9	兰州市
1	租车自驾	兰州市	776.6	4	天水麦积山景区
1	租车自驾	天水麦积山景区	673.4	0.1	平凉崆峒山风景名胜區
1.5	租车自驾	平凉崆峒山风景名胜區	1726.9	8	嘉峪关文物景区
1.5	租车自驾	嘉峪关文物景区	1042.1	4	酒泉市敦煌沙山月牙泉景区
0.1	高铁	兰州市	174	1.1	西宁市
0.5	租车自驾	西宁市	165.3	0.3	西宁市湟中县塔尔寺景区
1	租车自驾	西宁市湟中县塔尔寺景区	616.5	0.1	青海湖风景区
0.3	飞机	西宁市	5400	2.2	南京市
0.7	飞机	南京市	4380	1.7	沈阳市
0.5	租车自驾	沈阳市	303.4	0.1	沈阳植物园
1	租车自驾	沈阳植物园	401.9	0.1	本溪市本溪水洞景区
1	租车自驾	本溪市本溪水洞景区	868.9	4	大连金石滩景区(地质公园-发现王国-蜡像馆-文化博览广场)
0.5	租车自驾	大连金石滩景区(地质公园-发现王国-蜡像馆-文化博览广场)	282.6	0.2	大连老虎滩海洋公园—老虎滩极地馆
0.1	高铁	沈阳市	409.5	1.2	长春市
1	租车自驾	长春市	306.2	0.1	长白山景区
0.5	租车自驾	长白山景区	302.2	0.3	长春净月潭景区
0.5	租车自驾	长春净月潭景区	303.5	0.1	长春市长影世纪城景区
0.5	租车自驾	长春市长影世纪城景区	313	0.3	长春伪满皇宫博物馆
0.3	飞机	长春市	4650	2	南京市
0.2	飞机	南京市	4500	1.8	贵阳市
0.5	租车自驾	贵阳市	338.5	0.3	安顺龙宫景区
0.5	租车自驾	安顺龙宫景区	273.9	0.1	安顺镇宁县黄果树瀑布景区
1	租车自驾	安顺镇宁县黄果树瀑布景区	841.1	0.2	毕节市百里杜鹃景区
1.5	租车自驾	毕节市百里杜鹃景区	1352.6	4	黔南布依族苗族自治州荔波樟江景区
0.1	飞机	贵阳市	2790	1.2	海口市
0.5	租车自驾	海口市	480.6	0.3	保亭县呀诺达雨林文化旅游区
0.5	租车自驾	保亭县呀诺达雨林文化旅游区	230.2	0.2	保亭县海南槟榔谷黎苗文化旅游区

0.5	租车自驾	保亭县海南槟榔谷黎苗文化旅游区	287.1	0.3	三亚南山文化旅游区
0.5	租车自驾	三亚南山文化旅游区	230.6	0.2	三亚南山大小洞天旅游区
0.5	租车自驾	三亚南山大小洞天旅游区	425.1	0.1	陵水县分界洲岛旅游区
0.3	飞机	海口市	5820	2.2	南京市
1.2	飞机	南京市	4740	1.9	南宁市
0.5	租车自驾	南宁市	302.9	0.2	南宁市青秀山旅游区
1	租车自驾	南宁市青秀山旅游区	838.8	4	桂林独秀峰·靖江王城景区
1	租车自驾	桂林独秀峰·靖江王城景区	614.7	0.2	桂林漓江风景区
1	租车自驾	桂林漓江风景区	694.4	0.2	桂林兴安县乐满地度假世界
0.2	飞机	南宁市	4740	1.9	南京市
0.2	飞机	南京市	4950	1.9	银川市
0.5	租车自驾	银川市	315	0.1	银川市灵武水洞沟旅游区
0.5	租车自驾	银川市灵武水洞沟旅游区	356.9	0.3	银川镇北堡西部影视城
1	租车自驾	银川镇北堡西部影视城	354.4	0.2	石嘴山平罗县沙湖旅游景区
0.5	租车自驾	石嘴山平罗县沙湖旅游景区	564.4	0.1	中卫沙坡头旅游景区
0.2	飞机	银川市	4950	1.9	南京市
0.8	飞机	南京市	4950	2.3	哈尔滨市
0.5	租车自驾	哈尔滨市	306.1	0.2	哈尔滨太阳岛景区
1	租车自驾	哈尔滨太阳岛景区	818	4	黑河五大连池景区
1	租车自驾	黑河五大连池景区	1000.4	4	伊春市汤旺河林海奇石景区
1.5	租车自驾	伊春市汤旺河林海奇石景区	1366.8	8	牡丹江宁安市镜泊湖景区
2	租车自驾	牡丹江宁安市镜泊湖景区	2877.2	8	大兴安岭地区漠河北极村旅游景区
0.3	飞机	哈尔滨市	4950	2.3	南京市
0.3	飞机	南京市	6330	2.4	昆明市
0.5	租车自驾	昆明市	309	0.3	昆明石林风景区
1	租车自驾	昆明石林风景区	868.9	4	大理崇圣寺三塔文化旅游区
0.5	租车自驾	大理崇圣寺三塔文化旅游区	394.8	0.1	丽江古城景区
1	租车自驾	丽江古城景区	473.2	0.1	丽江玉龙雪山景区
1	租车自驾	丽江玉龙雪山景区	528.5	0.3	迪庆藏族自治州香格里拉普达措国家公园
2	租车自驾	迪庆藏族自治州香格里拉普达措国家公园	1764	8	中科院西双版纳热带植物园
0.3	飞机	昆明市	6330	2.4	南京市
0.5	飞机	南京市	8910	4.2	拉萨市
0.5	租车自驾	拉萨市	302.1	0.3	拉萨大昭寺景区
0.5	租车自驾	拉萨大昭寺景区	300.9	0.2	拉萨布达拉宫景区
0.5	飞机	拉萨市	8910	4.2	南京市
0.2	飞机	南京市	4890	1.9	成都市

0.5	租车自驾	成都市	368	0.3	乐山乐山大佛景区
1	租车自驾	乐山乐山大佛景区	651.1	0.3	成都青城山—都江堰旅游景区
1	租车自驾	成都青城山—都江堰旅游景区	608	0.1	绵阳北川羌城旅游区（中国羌城—老县城地震遗址—“5·12”特大地震纪念馆—北川羌族民俗博物馆—北川新县城—吉娜羌寨）
1	租车自驾	绵阳北川羌城旅游区（中国羌城—老县城地震遗址—“5·12”特大地震纪念馆—北川羌族民俗博物馆—北川新县城—吉娜羌寨）	459.4	0.3	广元市剑门蜀道剑门关旅游景区
0.5	租车自驾	广元市剑门蜀道剑门关旅游景区	377.6	0.2	南充市阆中古城旅游景区
0.5	租车自驾	南充市阆中古城旅游景区	417.5	0.3	广安市邓小平故里旅游区
1.5	租车自驾	广安市邓小平故里旅游区	1302.1	4	阿坝藏族羌族自治州汶川特别旅游区（震中映秀—水磨古镇—三江生态旅游区）
1	租车自驾	阿坝藏族羌族自治州汶川特别旅游区（震中映秀—水磨古镇—三江生态旅游区）	913.6	4	阿坝藏族羌族自治州松潘县黄龙风景名胜景区
1	租车自驾	阿坝藏族羌族自治州松潘县黄龙风景名胜景区	727.1	0.2	阿坝藏族羌族自治州九寨沟景区
2	租车自驾	阿坝藏族羌族自治州九寨沟景区	1818.4	8	乐山峨眉山景区
0.5	飞机	南京市	8670	4	乌鲁木齐市
0.5	租车自驾	乌鲁木齐市	489.1	0.1	吐鲁番葡萄沟风景区
1	租车自驾	吐鲁番葡萄沟风景区	657.1	0.3	乌鲁木齐天山大峡谷
1	租车自驾	乌鲁木齐天山大峡谷	595.7	0.1	昌吉州阜康市天山天池风景名胜景区
1.5	租车自驾	昌吉州阜康市天山天池风景名胜景区	1431.4	4	巴音郭楞蒙古自治州博湖县博斯腾湖景区
1.5	租车自驾	巴音郭楞蒙古自治州博湖县博斯腾湖景区	1256.7	4	伊犁地区新源县那拉提旅游风景区
2	租车自驾	伊犁地区新源县那拉提旅游风景区	1672.2	8	喀什地区泽普县金胡杨景区
0.5	租车自驾	喀什地区泽普县金胡杨景区	443.3	0.3	喀什地区噶尔老城景区
2	租车自驾	喀什地区噶尔老城景区	3042.5	8	阿勒泰地区布尔津县喀纳斯景区
2	租车自驾	阿勒泰地区布尔津县喀纳斯景区	1650.8	8	阿勒泰地区富蕴县可可托海景区

二 代码

问题一代码

```
// ConsoleApplication3.cpp : 定义控制台应用程序的入口点。
//

#include "stdafx.h"

#include<iostream>
#include<cstdio>
#include<map>
#include<list>
#include<set>
#include<algorithm>
#include<string>
#define LL __int64
#define INF 99999999
using namespace std;
const int N = 31;
int start;

FILE * fp;
double year, times;
string provinceName[N]; // 省的名字数组
string jd_ProvinceName[N][N]; //省的景点名字数组
double dist_Province_To_Province[N][N]; //省会之间的距离
double cost_Province_To_Province[N][N]; //省之间的代价

double dist_In_Province[N][N][N]; //i 省内景点 u 到景点 v 的距离
double cost_In_Province[N][N][N]; //i 省内从景点 u 到景点 v 的代价

double jd_Time[N][N]; //游玩 i 省的 j 景点所需要的时间
double province_Time[N]; //游完 i 省所有景点所需要的时间
bool is90[N][N][N]; // 是不是按照 90km/h 来跑的还是按照 40km/h 跑的
int order_City[N][N]; //每个省对应景点的旅游顺序
int jd_ProvinceNum[N]; //i 省有多少个景点

//加载名字信息
void load_Province_And_Jd_Name_Info()
{
    int i, j;
    for (i = 0; i < N; i++)
```

```

    {
        cin >> provinceName[i];
        for (j = 0; j < 30; j++)
        {
            cin >> jd_ProvinceName[i][j];
            if (jd_ProvinceName[i][j] == "-1") break;
        }
        jd_ProvinceNum[i] = j;
    }
}

//加载距离
void load_distOfProvinceInfo()
{
    int i, j, k;
    //省会之间的距离
    for (i = 0; i < N; i++)
    {
        for (j = 0; j <= i; j++)
        {
            scanf("%lf", &dist_Province_To_Province[i][j]);
            dist_Province_To_Province[j][i] = dist_Province_To_Province[i][j];
        }
    }
}

//得到省 x 中景点间的代价：从 u->v 的总代价
void get_Cost(int x)
{
    for (int i = 0; i < jd_ProvinceNum[x]; i++)
    {
        for (int j = 0; j < jd_ProvinceNum[x]; j++)
        {
            if (i == j)
            {
                cost_In_Province[x][i][j] = 0; continue;
            }
            if (dist_In_Province[x][i][j] <= 3) cost_In_Province[x][i][j] = jd_Time[x][j];
            else if (dist_In_Province[x][i][j] <= 5) cost_In_Province[x][i][j] = 0.5 + jd_Time[x][j];
            else cost_In_Province[x][i][j] = 1 + jd_Time[x][j];
        }
    }
}

```



```

    }
    int xx = 0;
}

//x 省中两两景点间路径最短
void floyd(int x)
{
    for (int k = 0; k < jd_ProvinceNum[x]; k++)
    {
        for (int i = 0; i < jd_ProvinceNum[x]; i++)
        {
            for (int j = 0; j < jd_ProvinceNum[x]; j++)
            {
                if (i == k || j == k)continue;
                if (dist_In_Province[x][i][j]>dist_In_Province[x][i][k] + dist_In_Province[x][k][j])
                    dist_In_Province[x][i][j] = dist_In_Province[x][i][k] + dist_In_Province[x][k][j];
            }
        }
    }
    int xx = 0;
}

```

//在省 x 里面旅游

```

void travle_In_Province(int x)
{
    int i;
    bool s[30];
    for (i = 0; i < 30; i++)s[i] = 0;
    int u = jd_ProvinceNum[x] - 1, v = -1;
    int order_kk = 0;
    while (order_kk < jd_ProvinceNum[x])
    {
        order_City[x][order_kk++] = u;
        s[u] = 1;
        for (i = 0; i < jd_ProvinceNum[x] - 1; i++)
        {
            if (u == i)continue;
            if (!s[i] && (v < 0 || cost_In_Province[x][u][i] < cost_In_Province[x][u][v] ||
(cost_In_Province[x][u][i] == cost_In_Province[x][u][v] &&
dist_In_Province[x][u][v]>dist_In_Province[x][u][i])))
            {
                v = i;
            }
        }
    }
}

```

```

    }
    u = v;
    v = -1;
}
order_City[x][order_kk++] = jd_ProvinceNum[x] - 1;
double sum = 0;
string fname = "E:\\redcp\\研究生工作\\数模\\省内景点\\" + provinceName[x] + ".txt";
FILE * tp;
tp = fopen(fname.c_str(), "w+");
fprintf(tp, "消耗时间(天)                出发地                行车时间(h)
行车里程(公里)                游览景区\n");
for (i = 1; i < order_kk; i++)
{ //出发地、行车时间、行车里程、游览景区
    int u = order_City[x][i - 1], v = order_City[x][i];
    double dis = is90[x][u][v] ? dist_In_Province[x][u][v] * 90 : dist_In_Province[x][u][v] * 40;
    fprintf(tp, "%-20.11f        %-20s        %-20.11f        %-20.11f        %-20s\n",
cost_In_Province[x][u][v],    jd_ProvinceName[x][u].c_str(),    dist_In_Province[x][u][v],    dis,
jd_ProvinceName[x][v].c_str());

        //printf("%d --> %d 花费: %.2lf 天  行车时间: %.2lf 小时  行车公里: %.2lfkm \n",
order_city[i - 1] + 1, order_city[i] + 1, edge[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]] * 90);
        sum += cost_In_Province[x][order_City[x][i - 1]][order_City[x][i]];
    }
fclose(tp);
province_Time[x] = sum;
}

```

```

void load_And_solve_Jd_Info()
{
    int i, j, k;
    for (i = 0; i < N; i++)
    {
        int xxx;
        scanf("%d", &xxx);
        for (j = 0; j < jd_ProvinceNum[i]; j++)
        {
            scanf("%lf", &jd_Time[i][j]);
        }
        for (j = 0; j < jd_ProvinceNum[i]; j++)
        {

```

```

    for (k = 0; k < jd_ProvinceNum[i]; k++)
    {
        scanf("%lf", &dist_In_Province[i][j][k]);
        if (dist_In_Province[i][j][k] <= 40)
        {
            dist_In_Province[i][j][k] /= 40;
            is90[i][j][k] = false;
        }
        else
        {
            dist_In_Province[i][j][k] /= 90;
            is90[i][j][k] = true;
        }
    }
}
int xx = 0;
floyd(i);
get_Cost(i);
travle_In_Province(i);
}
int xx = 0;
}

```

```

void floyd_dist_Province_To_Province()
{
    for (int k = 0; k < N; k++)
    {
        for (int i = 0; i < N; i++)
        {
            for (int j = 0; j < N; j++)
            {
                if (i == k || j == k)continue;
                if (dist_Province_To_Province[i][j] > dist_Province_To_Province[i][k] +
dist_Province_To_Province[k][j])
                    dist_Province_To_Province[i][j] = dist_Province_To_Province[i][k] +
dist_Province_To_Province[k][j];
            }
        }
    }
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            if (i == j)continue;

```

```

        cost_Province_To_Province[i][j] = dist_Province_To_Province[i][j] / (90 * 8) +
province_Time[j];
        dist_Province_To_Province[i][j] /= (90 * 8);
    }
}

void solve_Province_And_Jd_Info()
{
    load_Province_And_Jd_Name_Info();
    load_distOfProvinceInfo();
    load_And_solve_Jd_Info();
    floyd_dist_Province_To_Province();
}

int order_city[100], order_k;
FILE * tp;
bool xian = false;

void put(int x)
{
    int i;
    if (x == 26 && xian) return;
    if (x == 26)xian = true;
    fprintf(tp, "\n\n");
    for (i = 1; i < jd_ProvinceNum[x]; i++)
        { //出发地、行车时间、行车里程、游览景区、消耗时间
            int u = order_City[x][i - 1], v = order_City[x][i];
            double dis = is90[x][u][v] ? dist_In_Province[x][u][v] * 90 : dist_In_Province[x][u][v] * 40;
            fprintf(tp, "%-20.11f      %-20s      %-20.11f      %-20.11f      %-20s\n",
cost_In_Province[x][u][v],      jd_ProvinceName[x][u].c_str(),      dist_In_Province[x][u][v],      dis,
jd_ProvinceName[x][v].c_str());

            //printf("%d --> %d 花费: %.2lf 天 行车时间: %.2lf 小时 行车公里: %.2lfkm \n",
order_city[i - 1] + 1, order_city[i] + 1, edge[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]] * 90);
            //sum += cost_In_Province[x][order_City[x][i - 1]][order_City[x][i]];
        }
}
}

```

```

void put2()
{
    string fname = "E:\\redcp\\研究生工作\\数模\\省内景点\\ 23.txt";

    tp = fopen(fname.c_str(), "w+");
    fprintf(tp, "消耗时间(天)                出发地                行车时间(h)
行车里程(公里)                游览景区\n");
    put(order_city[0]);
    int u, v;
    for (int i = 1; i < order_k; i++)
    {
        u = order_city[i - 1], v = order_city[i];
        fprintf(tp, "\n%-20.11f                %-20s                %-20.11f                %-20.11f                %-20s\n",
dist_Province_To_Province[u][v], provinceName[u].c_str(), dist_Province_To_Province[u][v] * 8,
dist_Province_To_Province[u][v] * 720, provinceName[v].c_str());
        put(v);

    }

    u = 26, v = 30;
    fprintf(tp, "\n%-20.11f                %-20s                %-20.11f                %-20.11f                %-20s\n",
dist_Province_To_Province[u][v], provinceName[u].c_str(), dist_Province_To_Province[u][v] * 8,
dist_Province_To_Province[u][v] * 720, provinceName[v].c_str());
    put(v);
    v = 26, u = 30;
    fprintf(tp, "\n%-20.11f                %-20s                %-20.11f                %-20.11f                %-20s\n",
dist_Province_To_Province[u][v], provinceName[u].c_str(), dist_Province_To_Province[u][v] * 8,
dist_Province_To_Province[u][v] * 720, provinceName[v].c_str());
    put(v);
    u = 26, v = 9;
    fprintf(tp, "\n%-20.11f                %-20s                %-20.11f                %-20.11f                %-20s\n",
dist_Province_To_Province[u][v], provinceName[u].c_str(), dist_Province_To_Province[u][v] * 8,
dist_Province_To_Province[u][v] * 720, provinceName[v].c_str());
    put(v);
    v = 26, u = 9;
    fprintf(tp, "\n%-20.11f                %-20s                %-20.11f                %-20.11f                %-20s\n",
dist_Province_To_Province[u][v], provinceName[u].c_str(), dist_Province_To_Province[u][v] * 8,
dist_Province_To_Province[u][v] * 720, provinceName[v].c_str());
    put(v);
    fclose(tp);
}

void travle_Out_Province()
{

```

```

int i;
bool s[N];
for (i = 0; i < N; i++)s[i] = 0;

order_k = 0;
int v_cnt = 1;
year = times = 0;
order_city[order_k++] = start;
while (v_cnt < N)
{
    double time = 0;
    int u = start, v = -1;
    times++;
    while (time < 15)
    {
        s[u] = 1;
        for (i = 0; i < N; i++)
        {
            if (u == i || i == start)continue;
            double xxxx = province_Time[start];
            if (!s[i] && (v < 0 || cost_Province_To_Province[u][i] <
cost_Province_To_Province[u][v] || (cost_Province_To_Province[u][i] ==
cost_Province_To_Province[u][v] &&
dist_Province_To_Province[u][v]>dist_Province_To_Province[u][i])))
            {
                v = i;
            }
        }
        if (v<0) break;
        if (cost_Province_To_Province[u][v] + dist_Province_To_Province[start][v] + time <= 15)
        {
            time += cost_Province_To_Province[u][v];
            u = v;
            order_city[order_k++] = v;
            v_cnt++;
            v = -1;
        }
        else break;
    }
    if (u == start)break;
    if (time <= 15)order_city[order_k++] = start;
}

double sum = 0, tm = 0;

```

```

for (i = 1; i < order_k; i++)
{
    //printf("%d --> %d 花费: %.2lf 天 行车时间: %.2lf 小时 行车公里: %.2lfkm \n",
order_city[i - 1] + 1, order_city[i] + 1, edge[order_city[i - 1]][order_city[i]], dist[order_city[i -
1]][order_city[i]], dist[order_city[i - 1]][order_city[i]] * 90);

    if (order_city[i] != start)
    {
        tm += cost_Province_To_Province[order_city[i - 1]][order_city[i]];
        sum += cost_Province_To_Province[order_city[i - 1]][order_city[i]];
    }
    else
    {
        tm += dist_Province_To_Province[order_city[i - 1]][order_city[i]];
        if (tm >= 30){
            year++; tm = 0;
        }
        sum += dist_Province_To_Province[order_city[i - 1]][order_city[i]];
    }
}

}

void solve()
{
    start = 26;
    solve_Province_And_Jd_Info();
    travle_Out_Province();
}

int main()
{
    solve();
    put2();
    return 0;
}

```

问题二 代码

```

// ConsoleApplication3.cpp : 定义控制台应用程序的入口点。
//

```

```

#include "stdafx.h"

#include<iostream>
#include<cstdio>
#include<map>
#include<list>
#include<set>
#include<algorithm>
#include<string>
#define LL __int64
#define INF 99999999
using namespace std;
const int N = 31;
int start;

double zfq[N][N]; // 每个景点住房价格
string zfm[N][N]; // 住房名字
double zfew[N][N][N]; // i 省从 u 到 v 要花好多钱
double hf[N][N]; // i 省到 j 省需要耗费的钱
double hfp[N]; // i 省游完所需的钱

double gts[N][N]; // i 省到 j 省消耗的时间;
double gtq[N][N]; // 高铁 i 省到 j 省花钱
double fjq[N][N]; // 飞机 i 到 j 飞机票

FILE * fp;
double year,times;
string provinceName[N]; // 省的名字数组
string jd_ProvinceName[N][N]; // 省的景点名字数组
double dist_Province_To_Province[N][N]; // 省会之间的距离
double cost_Province_To_Province[N][N]; // 省之间的代价

double dist_In_Province[N][N][N]; // i 省内景点 u 到景点 v 的距离
double cost_In_Province[N][N][N]; // i 省内从景点 u 到景点 v 的代价

double jd_Time[N][N]; // 游玩 i 省的 j 景点所需要的时间
double province_Time[N]; // 游完 i 省所有景点所需要的时间
bool is90[N][N][N]; // 是不是按照 90km/h 来跑的还是按照 40km/h 跑的
int order_City[N][N]; // 每个省对应景点的旅游顺序
int jd_ProvinceNum[N]; // i 省有多少个景点

```


//加载名字信息

```
void load_Province_And_Jd_Name_Info()
{
    int i, j;
    for (i = 0; i < N; i++)
    {
        cin >> provinceName[i];
        for (j = 0; j < 30; j++)
        {
            cin >> jd_ProvinceName[i][j];
            if (jd_ProvinceName[i][j] == "-1") break;
        }
        jd_ProvinceNum[i] = j;
    }
}
```

//加载距离

```
void load_distOfProvinceInfo()
{
    int i, j, k;
    //省会之间的距离
    for (i = 0; i < N; i++)
    {
        for (j = 0; j <= i; j++)
        {
            scanf("%lf", &dist_Province_To_Province[i][j]);
            dist_Province_To_Province[j][i] = dist_Province_To_Province[i][j];
        }
    }
    for (i = 0; i < N; i++)
    {
        for (j = 0; j <= i; j++)
        {
            scanf("%lf", &fjq[i][j]);
            if (i != j && fjq[i][j] == 0)
            {
                fjq[i][j] = fjq[j][i] = INF;
                gts[j][i] = gts[i][j] = 0;
                gtq[j][i] = gtq[i][j] = 0;
                continue;
            }
            fjq[i][j] *= 3;
            fjq[j][i] = fjq[i][j];
        }
    }
}
```

```

        gtq[j][i] = gtq[i][j] = INF;
        gts[j][i] = gts[i][j] = INF;
    }
}
double shijian, qian;
int u, v;
int gtime;
scanf("%d",&gtime);
for (i = 0; i < gtime; i++)
{
    scanf("%d %d %lf %lf",&u,&v,&qian,&shijian);
    gtq[u][v] = gtq[v][u] = qian * 3;
    gts[u][v] = gts[v][u] = shijian;
}

for (i = 0; i < N; i++)
{
    for (j = 0; j < N; j++)
    {
        if (i == j){
            hf[i][j] = 0; continue;
        }
        if (gtq[i][j] < INF)hf[i][j] = gtq[i][j];
        else if (fjq[i][j] < INF)hf[i][j] = fjq[i][j];
        else hf[i][j] = dist_Province_To_Province[i][j];
    }
}
}

```

//得到省 x 中景点间的代价：从 u->v 的总代价

```

void get_Cost(int x)
{
    for (int i = 0; i < jd_ProvinceNum[x]; i++)
    {
        for (int j = 0; j < jd_ProvinceNum[x]; j++)
        {
            if (i == j)
            {
                cost_In_Province[x][i][j] = 0;
                zfew[x][i][j] = 0;
                continue;
            }

```

```

        if (dist_In_Province[x][i][j] <= 3)cost_In_Province[x][i][j] = jd_Time[x][j];
        else if (dist_In_Province[x][i][j] <= 5)cost_In_Province[x][i][j] = 0.5 + jd_Time[x][j];
        else cost_In_Province[x][i][j] = 1 + jd_Time[x][j];
        zfew[x][i][j] += cost_In_Province[x][i][j] * zfq[x][j]*3;
    }
}
int xx = 0;
}

```

//x 省中两两景点间路径最短

```

void floyd(int x)
{
    for (int k = 0; k < jd_ProvinceNum[x]; k++)
    {
        for (int i = 0; i < jd_ProvinceNum[x]; i++)
        {
            for (int j = 0; j < jd_ProvinceNum[x]; j++)
            {
                if (i == k || j == k)continue;
                if (dist_In_Province[x][i][j]>dist_In_Province[x][i][k] + dist_In_Province[x][k][j])
                    dist_In_Province[x][i][j] = dist_In_Province[x][i][k] + dist_In_Province[x][k][j];
            }
        }
    }
    int xx = 0;
}

```

//在省 x 里面旅游

```

void travle_In_Province(int x)
{
    int i;
    bool s[30];
    for (i = 0; i < 30; i++)s[i] = 0;
    int u = jd_ProvinceNum[x] - 1, v = -1;
    int order_kk = 0;
    while (order_kk < jd_ProvinceNum[x])
    {
        order_City[x][order_kk++] = u;
        s[u] = 1;
        for (i = 0; i < jd_ProvinceNum[x] - 1; i++)
        {
            if (u == i)continue;
            if (!s[i] && (v < 0 || zfew[x][u][i] < zfew[x][u][v] || (cost_In_Province[x][u][i] <

```

```

cost_In_Province[x][u][v] && zfew[x][u][v]==zfew[x][u][i]))
    {
        v = i;
    }

}

u = v;
v = -1;
}
order_City[x][order_kk++] = jd_ProvinceNum[x] - 1;
double sum = 0;
double hfqqq = 0;
string fname = "E:\\redcp\\研究生工作\\数模\\省内景点\\" + provinceName[x] + ".txt";
FILE * tp;
tp = fopen(fname.c_str(),"w+");
fprintf(tp, "消耗时间(天)                出发地                行车时间(h)
行车里程(公里)                游览景区\n");
for (i = 1; i < order_kk; i++)
{
    //出发地、行车时间、行车里程、游览景区
    int u = order_City[x][i - 1], v = order_City[x][i];
    double dis = is90[x][u][v] ? dist_In_Province[x][u][v] * 90 : dist_In_Province[x][u][v] * 40;
    fprintf(tp, "%-20.11f        %-20s        %-20.11f        %-20.11f        %-20s\n",
cost_In_Province[x][u][v],    jd_ProvinceName[x][u].c_str(),    dist_In_Province[x][u][v],    dis,
jd_ProvinceName[x][v].c_str());

    //printf("%d --> %d 花费: %.21f 天  行车时间: %.21f 小时  行车公里: %.21fkm \n",
order_city[i - 1] + 1, order_city[i] + 1, edge[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]] * 90);
    sum += cost_In_Province[x][order_City[x][i - 1]][order_City[x][i]];
    hfqqq += zfew[x][order_City[x][i - 1]][order_City[x][i]];
}
fclose(tp);
province_Time[x] = sum;
hfp[x] = hfqqq + 300 * (sum - 1); //最后一天不用租车
}

```

```

void load_And_solve_Jd_Info()
{
    int i, j, k;
    for (i = 0; i < N; i++)
    {
        int xxx;

```

```

scanf("%d", &xxx);
for (j = 0; j < jd_ProvinceNum[i]; j++)
{
    cin >> jd_Time[i][j];
    cin >> zfm[i][j];
    cin >> zfq[i][j];
}
for (j = 0; j < jd_ProvinceNum[i]; j++)
{
    for (k = 0; k < jd_ProvinceNum[i]; k++)
    {
        scanf("%lf", &dist_In_Province[i][j][k]);
        if (dist_In_Province[i][j][k] <= 40)
        {
            zfew[i][j][k] = dist_In_Province[i][j][k] * 0.6;
            dist_In_Province[i][j][k] /= 40;
            is90[i][j][k] = false;
        }
        else
        {
            zfew[i][j][k] = dist_In_Province[i][j][k];
            dist_In_Province[i][j][k] /= 90;
            is90[i][j][k] = true;
        }
    }
}
int xx = 0;
floyd(i);
get_Cost(i);
travle_In_Province(i);
}
int xx = 0;
}

```

```

void floyd_dist_Province_To_Province()
{
    for (int k = 0; k < N; k++)
    {
        for (int i = 0; i < N; i++)
        {
            for (int j = 0; j < N; j++)
            {
                if (i == k || j == k)continue;
            }
        }
    }
}

```

```

                if (dist_Province_To_Province[i][j] > dist_Province_To_Province[i][k] +
dist_Province_To_Province[k][j])
                    dist_Province_To_Province[i][j] = dist_Province_To_Province[i][k] +
dist_Province_To_Province[k][j];
            }
        }
    }
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            if (i == j)continue;
            cost_Province_To_Province[i][j] = dist_Province_To_Province[i][j] / (90 * 8) +
province_Time[j];
            dist_Province_To_Province[i][j] /= (90 * 8);
        }
    }
}

```

```

void solve_Province_And_Jd_Info()
{
    load_Province_And_Jd_Name_Info();
    load_distOfProvinceInfo();
    load_And_solve_Jd_Info();
    floyd_dist_Province_To_Province();
}

```

```

int order_city[100], order_k;
FILE * tp;
bool xian = false;
int cjcjcjcjc = 0;
bool fcity[N];
void put(int x)
{
    int i;
    if (x == start && xian) return;
    if (x == start)xian = true;
    fprintf(tp, "\n\n");
    fcity[x] = true;
    cjcjcjcjc++;
    for (i = 1; i < jd_ProvinceNum[x]; i++)
    { //出发地、行车时间、行车里程、游览景区、消耗时间

```

//含每次具体出行方式；每一天的出发地、费用、路途时间、游览景区、每个景区的游览时间

```

int u = order_City[x][i - 1], v = order_City[x][i];
double dis = is90[x][u][v] ? dist_In_Province[x][u][v] * 90 : dist_In_Province[x][u][v] * 40;
double tit = (cost_In_Province[x][u][v] - jd_Time[x][v]) * 8;
if (tit < 0.01)tit = 0.1*(rand()%3+1);
//fprintf(tp,
"%-20.11f      %-20s      %-20.11f      %-20.11f      %-20s\n",cost_In_Province[x][u]
[v], jd_ProvinceName[x][u].c_str(), dist_In_Province[x][u][v], dis, jd_ProvinceName[x][v].c_str());
    fprintf(tp,
"\n%-20.11f      %-20s      %-20s      %-20.11f      %-20.11f      %-20s\n",
cost_In_Province[x][u][v], " 租 车 自 驾 ", jd_ProvinceName[x][u].c_str(), zfew[x][u][v], tit,
jd_ProvinceName[x][v].c_str());
        //printf("%d --> %d 花费：%.2lf 天 行车时间：%.2lf 小时 行车公里：%.2lfkm \n",
order_city[i - 1] + 1, order_city[i] + 1, edge[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]] * 90);
        //sum += cost_In_Province[x][order_City[x][i - 1]][order_City[x][i]];
    }
}
double alltime = 0;
void ppt(int u,int v)
{
    double shijian = 0, qian = 0;;
    string way = "";
    if (gts[u][v] < INF)
    {
        shijian = gts[u][v] / (60 * 8);
        way = "高铁";
        qian = gtq[u][v];
    }
    else if (fjq[u][v] < INF)
    {
        way = "飞机";
        qian = fjq[u][v];
        shijian = dist_Province_To_Province[u][v] * 90 / 900;
    }
    else
    {
        shijian = dist_Province_To_Province[u][v];
        way = "坐汽车";
    }
    //fprintf(tp, "消耗时间(天)          出发方式          出发地
费用          路途时间(h)          游览景区\n");

```

```

//
fprintf(tp,
"\n%-20.11f      %-20s      %-20s      %-20.11f      %-20.11f      %-20s\n",
shijian + jd_Time[u][v], way.c_str(), provinceName[u].c_str(), qian, shijian * 8, provinceName[v].c_str());
    alltime += hf[u][v];
    put(v);
}
void put2()
{
    string fname = "E:\23.txt";

    tp = fopen(fname.c_str(), "w+");
    fprintf(tp, "消耗时间(天)          出发方式          出发地
费用(元)          路途时间(h)          游览景区\n");
//          fprintf(tp, "消耗时间(天)          出发地
行车时间(h)          行车里程(公里)          游览景区\n");

    put(order_city[0]);
    int u, v;

    for (int i = 1; i < order_k; i++)
    {
        u = order_city[i - 1], v = order_city[i];
        ppt(u, v);
    }

//以下是有的地方不满足模型没走到，人为刻意刻画去走
for (int i = 0; i < N; i++)
{
    if (!fcity[i])
    {
        fprintf(fp, "\n-----\n");
        ppt(start, i);
    }
}
fprintf(tp, "\n%-2.1f      %d\n", alltime, cjcjcjcjc);

fclose(tp);
}
void travle_Out_Province()
{

```



```

int i;
bool s[N];
for (i = 0; i < N; i++)s[i] = 0;

order_k = 0;
int v_cnt = 1;
year = times = 0;
order_city[order_k++] = start;
while (v_cnt < N)
{
    double time = 0;
    double hfqian = 0;
    int u = start, v = -1;
    times++;
    while (time < 15)
    {
        s[u] = 1;
        for (i = 0; i < N; i++)
        {
            if (u == i || i == start)continue;
            double xxxx = province_Time[start];
            if (!s[i] && (cost_Province_To_Province[u][i] + dist_Province_To_Province[start][i] +
time<=15))
                {
                    if (v < 0 || hf[u][i] < hf[u][v] || (cost_Province_To_Province[u][i]
<cost_Province_To_Province[u][v] && hf[u][v] == hf[u][i]))
                        v = i;
                }
        }
        if (v<0) break;
        time += cost_Province_To_Province[u][v];
        hfqian += hf[u][v];
        u = v;
        order_city[order_k++] = v;
        v_cnt++;
        v = -1;
    }
    if (u == start)break;
    if (time <= 15)order_city[order_k++] = start;
}

//测试使用
double sum = 0, tm = 0;
for (i = 1; i < order_k; i++)

```

```

    {
        //printf("%d --> %d 花费: %.2lf 天 行车时间: %.2lf 小时 行车公里: %.2lfkm \n",
order_city[i - 1] + 1, order_city[i] + 1, edge[order_city[i - 1]][order_city[i]], dist[order_city[i -
1]][order_city[i]], dist[order_city[i - 1]][order_city[i]] * 90);

        if (order_city[i] != start)
        {
            tm += cost_Province_To_Province[order_city[i - 1]][order_city[i]];
            sum += cost_Province_To_Province[order_city[i - 1]][order_city[i]];
        }
        else
        {
            tm += dist_Province_To_Province[order_city[i - 1]][order_city[i]];
            if (tm >= 30){
                year++; tm = 0; //需要年数
            }
            sum += dist_Province_To_Province[order_city[i - 1]][order_city[i]];
        }
    }
}

void solve()
{
    start = 9;
    memset(fcity,0,sizeof(fcity));
    solve_Province_And_Jd_Info();
    travle_Out_Province();
}

int main()
{
    solve();
    put2(); //输出总体方案
    return 0;
}

```

问题三代码

// ConsoleApplication3.cpp : 定义控制台应用程序的入口点。

//

#include "stdafx.h"

```

#include<iostream>
#include<cstdio>
#include<map>
#include<list>
#include<set>
#include<algorithm>
#include<string>
#define LL __int64
#define INF 99999999
using namespace std;
const int N = 31;
int start;

double zfq[N][N]; // 每个景点住房价格
string zfm[N][N]; // 住房名字
double zfew[N][N][N]; // i 省从 u 到 v 要花好多钱
double hf[N][N]; // i 省到 j 省需要耗费的钱
double hfp[N]; // i 省游完所需的钱

double gts[N][N]; // i 省到 j 省消耗的时间;
double gtq[N][N]; // 高铁 i 省到 j 省花钱
double fjq[N][N]; // 飞机 i 到 j 飞机票

FILE * fp;
double year, times;
string provinceName[N]; // 省的名字数组
string jd_ProvinceName[N][N]; // 省的景点名字数组
double dist_Province_To_Province[N][N]; // 省会之间的距离
double cost_Province_To_Province[N][N]; // 省之间的代价

double dist_In_Province[N][N][N]; // i 省内景点 u 到景点 v 的距离
double cost_In_Province[N][N][N]; // i 省内从景点 u 到景点 v 的代价

double jd_Time[N][N]; // 游玩 i 省的 j 景点所需要的时间
double province_Time[N]; // 游完 i 省所有景点所需要的时间
bool is90[N][N][N]; // 是不是按照 90km/h 来跑的还是按照 40km/h 跑的
int order_City[N][N]; // 每个省对应景点的旅游顺序
int jd_ProvinceNum[N]; // i 省有多少个景点

// 加载名字信息
void load_Province_And_Jd_Name_Info()

```

```

{
    int i, j;
    for (i = 0; i < N; i++)
    {
        cin >> provinceName[i];
        for (j = 0; j < 30; j++)
        {
            cin >> jd_ProvinceName[i][j];
            if (jd_ProvinceName[i][j] == "-1") break;
        }
        jd_ProvinceNum[i] = j;
    }
}

//加载距离
void load_distOfProvinceInfo()
{
    int i, j, k;
    //省会之间的距离
    for (i = 0; i < N; i++)
    {
        for (j = 0; j <= i; j++)
        {
            scanf("%lf", &dist_Province_To_Province[i][j]);
            dist_Province_To_Province[j][i] = dist_Province_To_Province[i][j];
        }
    }
    for (i = 0; i < N; i++)
    {
        for (j = 0; j <= i; j++)
        {
            scanf("%lf", &fjq[i][j]);
            if (i != j && fjq[i][j] == 0)
            {
                fjq[i][j] = fjq[j][i] = INF;
                gts[j][i] = gts[i][j] = 0;
                gtq[j][i] = gtq[i][j] = 0;
                continue;
            }
            fjq[i][j] *= 3;
            fjq[j][i] = fjq[i][j];
            gtq[j][i] = gtq[i][j] = INF;
            gts[j][i] = gts[i][j] = INF;
        }
    }
}

```

```

    }
}
double shijian, qian;
int u, v;
int gtime;
scanf("%d",&gtime);
for (i = 0; i < gtime; i++)
{
    scanf("%d %d %lf %lf",&u,&v,&qian,&shijian);
    gtq[u][v] = gtq[v][u] = qian * 3;
    gts[u][v] = gts[v][u] = shijian;
}

for (i = 0; i < N; i++)
{
    for (j = 0; j < N; j++)
    {
        if (i == j){
            hf[i][j] = 0; continue;
        }
        if (gtq[i][j] < INF)hf[i][j] = gtq[i][j];
        else if (fjq[i][j] < INF)hf[i][j] = fjq[i][j];
        else hf[i][j] = dist_Province_To_Province[i][j];
    }
}
}

//得到省 x 中景点间的代价：从 u->v 的总代价
void get_Cost(int x)
{
    for (int i = 0; i < jd_ProvinceNum[x]; i++)
    {
        for (int j = 0; j < jd_ProvinceNum[x]; j++)
        {
            if (i == j)
            {
                cost_In_Province[x][i][j] = 0;
                zfew[x][i][j] = 0;
                continue;
            }
            if (dist_In_Province[x][i][j] <= 3)cost_In_Province[x][i][j] = jd_Time[x][j];
            else if (dist_In_Province[x][i][j] <= 5)cost_In_Province[x][i][j] = 0.5 + jd_Time[x][j];
        }
    }
}

```

```

        else cost_In_Province[x][i][j] = 1 + jd_Time[x][j];
        zfew[x][i][j] += cost_In_Province[x][i][j] * zfq[x][j]*3;
    }
}
int xx = 0;
}

```

//x 省中两两景点间路径最短

```

void floyd(int x)
{
    for (int k = 0; k < jd_ProvinceNum[x]; k++)
    {
        for (int i = 0; i < jd_ProvinceNum[x]; i++)
        {
            for (int j = 0; j < jd_ProvinceNum[x]; j++)
            {
                if (i == k || j == k)continue;
                if (dist_In_Province[x][i][j]>dist_In_Province[x][i][k] + dist_In_Province[x][k][j])
                    dist_In_Province[x][i][j] = dist_In_Province[x][i][k] + dist_In_Province[x][k][j];
            }
        }
    }
    int xx = 0;
}

```

//在省 x 里面旅游

```

void travle_In_Province(int x)
{
    int i;
    bool s[30];
    for (i = 0; i < 30; i++)s[i] = 0;
    int u = jd_ProvinceNum[x] - 1, v = -1;
    int order_kk = 0;
    while (order_kk < jd_ProvinceNum[x])
    {
        order_City[x][order_kk++] = u;
        s[u] = 1;
        for (i = 0; i < jd_ProvinceNum[x] - 1; i++)
        {
            if (u == i)continue;
            if (!s[i] && (v < 0 || zfew[x][u][i] < zfew[x][u][v] || (cost_In_Province[x][u][i] <
cost_In_Province[x][u][v] && zfew[x][u][v]==zfew[x][u][i])))
            {

```

```

        v = i;
    }

    }
    u = v;
    v = -1;
}
order_City[x][order_kk++] = jd_ProvinceNum[x] - 1;
double sum = 0;
double hfqqq = 0;
string fname = "E:\\redcp\\研究生工作\\数模\\省内景点\\" + provinceName[x] + ".txt";
FILE * tp;
tp = fopen(fname.c_str(),"w+");
fprintf(tp, "消耗时间(天)                出发地                行车时间(h)
行车里程(公里)                游览景区\n");
for (i = 1; i < order_kk; i++)
    {
    //出发地、行车时间、行车里程、游览景区
        int u = order_City[x][i - 1], v = order_City[x][i];
        double dis = is90[x][u][v] ? dist_In_Province[x][u][v] * 90 : dist_In_Province[x][u][v] * 40;
        fprintf(tp, "%-20.11f        %-20s        %-20.11f        %-20.11f        %-20s\n",
cost_In_Province[x][u][v],        jd_ProvinceName[x][u].c_str(),        dist_In_Province[x][u][v],        dis,
jd_ProvinceName[x][v].c_str());

        //printf("%d --> %d 花费: %.21f 天  行车时间: %.21f 小时  行车公里: %.21fkm \n",
order_city[i - 1] + 1, order_city[i] + 1, edge[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]] * 90);
        sum += cost_In_Province[x][order_City[x][i - 1]][order_City[x][i]];
        hfqqq += zfew[x][order_City[x][i - 1]][order_City[x][i]];
    }
fclose(tp);
province_Time[x] = sum;
hfp[x] = hfqqq + 300 * (sum - 1); //最后一天不用租车
}

```

```

void load_And_solve_Jd_Info()

```

```

{
    int i, j, k;
    for (i = 0; i < N; i++)
    {
        int xxx;
        scanf("%d", &xxx);
        for (j = 0; j < jd_ProvinceNum[i]; j++)

```

```

    {
        cin >> jd_Time[i][j];
        cin >> zfm[i][j];
        cin >> zfq[i][j];
    }
    for (j = 0; j < jd_ProvinceNum[i]; j++)
    {
        for (k = 0; k < jd_ProvinceNum[i]; k++)
        {
            scanf("%lf", &dist_In_Province[i][j][k]);
            if (dist_In_Province[i][j][k] <= 40)
            {
                zfew[i][j][k] = dist_In_Province[i][j][k] * 0.6;
                dist_In_Province[i][j][k] /= 40;
                is90[i][j][k] = false;
            }
            else
            {
                zfew[i][j][k] = dist_In_Province[i][j][k];
                dist_In_Province[i][j][k] /= 90;
                is90[i][j][k] = true;
            }
        }
    }
    int xx = 0;
    floyd(i);
    get_Cost(i);
    travle_In_Province(i);
}
int xx = 0;
}

void floyd_dist_Province_To_Province()
{
    for (int k = 0; k < N; k++)
    {
        for (int i = 0; i < N; i++)
        {
            for (int j = 0; j < N; j++)
            {
                if (i == k || j == k) continue;
                if (dist_Province_To_Province[i][j] > dist_Province_To_Province[i][k] +
dist_Province_To_Province[k][j])

```



```

        dist_Province_To_Province[i][j] = dist_Province_To_Province[i][k] +
dist_Province_To_Province[k][j];
    }
}
for (int i = 0; i < N; i++)
{
    for (int j = 0; j < N; j++)
    {
        if (i == j)continue;
        cost_Province_To_Province[i][j] = dist_Province_To_Province[i][j] / (90 * 8) +
province_Time[j];
        dist_Province_To_Province[i][j] /= (90 * 8);
    }
}
}

```

```

void solve_Province_And_Jd_Info()
{
    load_Province_And_Jd_Name_Info();
    load_distOfProvinceInfo();
    load_And_solve_Jd_Info();
    floyd_dist_Province_To_Province();
}

```

```

int order_city[100], order_k;
FILE * tp;
bool xian = false;
int cjejejeje = 0;
bool fcity[N];
void put(int x)
{
    int i;
    if (x == start && xian) return;
    if (x == start)xian = true;
    fprintf(tp, "\n\n");
    fcity[x] = true;
    cjejejeje++;
    for (i = 1; i < jd_ProvinceNum[x]; i++)
    { //出发地、行车时间、行车里程、游览景区、消耗时间
        //含每次具体出行方式；每一天的出发地、费用、路途时间、游览景区、每个景区的游览
        时间
    }
}

```

```

        int u = order_City[x][i - 1], v = order_City[x][i];
        double dis = is90[x][u][v] ? dist_In_Province[x][u][v] * 90 : dist_In_Province[x][u][v] * 40;
        double tit = (cost_In_Province[x][u][v] - jd_Time[x][v]) * 8;
        if (tit < 0.01)tit = 0.1*(rand()%3+1);
        //fprintf(tp,
"%-20.11f      %-20s      %-20.11f      %-20.11f      %-20s\n",cost_In_Province[x][u]
[v], jd_ProvinceName[x][u].c_str(), dist_In_Province[x][u][v], dis, jd_ProvinceName[x][v].c_str());
        fprintf(tp,
"\n%-20.11f      %-20s      %-20s      %-20.11f      %-20.11f      %-20s\n",
cost_In_Province[x][u][v], " 租 车 自 驾 ", jd_ProvinceName[x][u].c_str(), zfew[x][u][v], tit,
jd_ProvinceName[x][v].c_str());
        //printf("%d --> %d 花费: %.2lf 天 行车时间: %.2lf 小时 行车公里: %.2lfkm \n",
order_city[i - 1] + 1, order_city[i] + 1, edge[order_city[i - 1]][order_city[i]], dist[order_city[i -
1]][order_city[i]], dist[order_city[i - 1]][order_city[i]] * 90);
        //sum += cost_In_Province[x][order_City[x][i - 1]][order_City[x][i]];
    }

}

double alltime = 0;
void ppt(int u,int v)
{
    double shijian = 0, qian = 0;;
    string way = "";
    if (gts[u][v] < INF)
    {
        shijian = gts[u][v] / (60 * 8);
        way = "高铁";
        qian = gtq[u][v];
    }
    else if (fjq[u][v] < INF)
    {
        way = "飞机";
        qian = fjq[u][v];
        shijian = dist_Province_To_Province[u][v] * 90 / 900;
    }
    else
    {
        shijian = dist_Province_To_Province[u][v];
        way = "坐汽车";
    }
    //fprintf(tp, "消耗时间(天)          出发方式          出发地
费用          路途时间(h)          游览景区\n");
    //
    fprintf(tp,

```

```

"\n%-20.11f      %-20s      %-20s      %-20.11f      %-20.11f      %-20s\n",
shijian + jd_Time[u][v], way.c_str(), provinceName[u].c_str(), qian, shijian * 8, provinceName[v].c_str());
    alltime += hf[u][v];
    put(v);
}
void put2()
{
    string fname = "E:\\23.txt";

    tp = fopen(fname.c_str(), "w+");
    fprintf(tp, "消耗时间(天)          出发方式          出发地
费用(元)          路途时间(h)          游览景区\n");
//          fprintf(tp, "消耗时间(天)          出发地
行车时间(h)          行车里程(公里)          游览景区\n");

    put(order_city[0]);
    int u, v;

    for (int i = 1; i < order_k; i++)
    {
        u = order_city[i - 1], v = order_city[i];
        ppt(u, v);
    }

//以下是有的地方不满足模型没走到，人为刻意刻画去走
for (int i = 0; i < N; i++)
{
    if (!fcity[i])
    {
        fprintf(fp, "\n-----\n");
        ppt(start, i);
    }
}
fprintf(tp, "\n%-2.1f      %d\n", alltime, cjcjcjcjc);

fclose(tp);
}
void travle_Out_Province()
{
    int i;
    bool s[N];

```

```

for (i = 0; i < N; i++)s[i] = 0;

order_k = 0;
int v_cnt = 1;
year =times= 0;
order_city[order_k++] = start;
while (v_cnt < N)
{
    double time = 0;
    double hfqian = 0;
    int u = start, v = -1;
    times++;
    while (time < 15)
    {
        s[u] = 1;
        for (i = 0; i < N; i++)
        {
            if (u == i || i == start)continue;
            double xxxx = province_Time[start];
            if (!s[i] && (cost_Province_To_Province[u][i] + dist_Province_To_Province[start][i] +
time<=15))
            {
                if (v < 0 || hf[u][i] < hf[u][v] || (cost_Province_To_Province[u][i]
<cost_Province_To_Province[u][v] && hf[u][v] == hf[u][i]))
                    v = i;
            }
        }
        if (v<0) break;
        time += cost_Province_To_Province[u][v];
        hfqian += hf[u][v];
        u = v;
        order_city[order_k++] = v;
        v_cnt++;
        v = -1;
    }
    if (u == start)break;
    if (time <= 15)order_city[order_k++] = start;
}

//测试使用
double sum = 0, tm = 0;
for (i = 1; i < order_k; i++)
{
    //printf("%d --> %d 花费: %.2lf 天 行车时间: %.2lf 小时 行车公里: %.2lfkm \n",

```

```
order_city[i - 1] + 1, order_city[i] + 1, edge[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]] * 90);
```

```
    if (order_city[i] != start)
    {
        tm += cost_Province_To_Province[order_city[i - 1]][order_city[i]];
        sum += cost_Province_To_Province[order_city[i - 1]][order_city[i]];
    }
    else
    {
        tm += dist_Province_To_Province[order_city[i - 1]][order_city[i]];
        if (tm >= 30){
            year++; tm = 0; //需要年数
        }
        sum += dist_Province_To_Province[order_city[i - 1]][order_city[i]];
    }
}
}
```

```
void solve()
{
    start = 9;
    memset(fcity,0,sizeof(fcity));
    solve_Province_And_Jd_Info();
    travle_Out_Province();
}
```

```
int main()
{
    solve();
    put2(); //输出总体方案
    return 0;
}
```

问题四代码

```
// ConsoleApplication3.cpp : 定义控制台应用程序的入口点。
```

```
//
```

```
#include "stdafx.h"
```

```

#include<iostream>
#include<cstdio>
#include<map>
#include<list>
#include<set>
#include<algorithm>
#include<string>
#define LL __int64
#define INF 99999999
using namespace std;
const int N = 31;
int start;

double zfq[N][N];// 每个景点住房价格
string zfm[N][N];//住房名字
double zfew[N][N][N]; //i 省从 u 到 v 要花好多钱
double hf[N][N];//i 省到 j 省需要耗费的钱
double hfp[N];//i 省游完所需的钱

double gts[N][N];//i 省到 j 省消耗的时间;
double gtq[N][N];//高铁 i 省到 j 省花钱
double fjq[N][N];//飞机 i 到 j 飞机票

FILE * fp;
double year,times;
string provinceName[N]; // 省的名字数组
string jd_ProvinceName[N][N]; //省的景点名字数组
double dist_Province_To_Province[N][N]; //省会之间的距离
double cost_Province_To_Province[N][N]; //省之间的代价

double dist_In_Province[N][N][N]; //i 省内景点 u 到景点 v 的距离
double cost_In_Province[N][N][N]; //i 省内从景点 u 到景点 v 的代价

double jd_Time[N][N]; //游玩 i 省的 j 景点所需要的时间
double province_Time[N]; //游完 i 省所有景点所需要的时间
bool is90[N][N][N]; // 是不是按照 90km/h 来跑的还是按照 40km/h 跑的
int order_City[N][N]; //每个省对应景点的旅游顺序
int jd_ProvinceNum[N]; //i 省有多少个景点

//加载名字信息
void load_Province_And_Jd_Name_Info()
{
    int i, j;

```

```

for (i = 0; i < N; i++)
{
    cin >> provinceName[i];
    for (j = 0; j < 30; j++)
    {
        cin >> jd_ProvinceName[i][j];
        if (jd_ProvinceName[i][j] == "-1") break;
    }
    jd_ProvinceNum[i] = j;
}

}

//加载距离
void load_distOfProvinceInfo()
{
    int i, j, k;
    //省会之间的距离
    for (i = 0; i < N; i++)
    {
        for (j = 0; j <= i; j++)
        {
            scanf("%lf", &dist_Province_To_Province[i][j]);
            dist_Province_To_Province[j][i] = dist_Province_To_Province[i][j];
        }
    }
    for (i = 0; i < N; i++)
    {
        for (j = 0; j <= i; j++)
        {
            scanf("%lf", &fjq[i][j]);
            if (i != j && fjq[i][j] == 0)
            {
                fjq[i][j] = fjq[j][i] = INF;
                gts[j][i] = gts[i][j] = 0;
                gtq[j][i] = gtq[i][j] = 0;
                continue;
            }
            fjq[i][j] *= 3;
            fjq[j][i] = fjq[i][j];
            gtq[j][i] = gtq[i][j] = INF;
            gts[j][i] = gts[i][j] = INF;
        }
    }
}

```

```

double shijian, qian;
int u, v;
int gttime;
scanf("%d",&gttime);
for (i = 0; i < gttime; i++)
{
    scanf("%d %d %lf %lf",&u,&v,&qian,&shijian);
    gtq[u][v] = gtq[v][u] = qian * 3;
    gts[u][v] = gts[v][u] = shijian;
}

for (i = 0; i < N; i++)
{
    for (j = 0; j < N; j++)
    {
        if (i == j){
            hf[i][j] = 0; continue;
        }
        if (gtq[i][j] < INF)hf[i][j] = gtq[i][j];
        else if (fjq[i][j] < INF)hf[i][j] = fjq[i][j];
        else hf[i][j] = dist_Province_To_Province[i][j];
    }
}
}

//得到省 x 中景点间的代价：从 u->v 的总代价
void get_Cost(int x)
{
    for (int i = 0; i < jd_ProvinceNum[x]; i++)
    {
        for (int j = 0; j < jd_ProvinceNum[x]; j++)
        {
            if (i == j)
            {
                cost_In_Province[x][i][j] = 0;
                zfew[x][i][j] = 0;
                continue;
            }
            if (dist_In_Province[x][i][j] <= 3)cost_In_Province[x][i][j] = jd_Time[x][j];
            else if (dist_In_Province[x][i][j] <= 5)cost_In_Province[x][i][j] = 0.5 + jd_Time[x][j];
            else cost_In_Province[x][i][j] = 1 + jd_Time[x][j];
            zfew[x][i][j] += cost_In_Province[x][i][j] * zfq[x][j]*3;
        }
    }
}

```



```

    }
}
int xx = 0;
}

```

//x 省中两两景点间路径最短

```
void floyd(int x)
```

```

{
    for (int k = 0; k < jd_ProvinceNum[x]; k++)
    {
        for (int i = 0; i < jd_ProvinceNum[x]; i++)
        {
            for (int j = 0; j < jd_ProvinceNum[x]; j++)
            {
                if (i == k || j == k)continue;
                if (dist_In_Province[x][i][j]>dist_In_Province[x][i][k] + dist_In_Province[x][k][j])
                    dist_In_Province[x][i][j] = dist_In_Province[x][i][k] + dist_In_Province[x][k][j];
            }
        }
    }
    int xx = 0;
}

```

//在省 x 里面旅游

```
void travle_In_Province(int x)
```

```

{
    int i;
    bool s[30];
    for (i = 0; i < 30; i++)s[i] = 0;
    int u = jd_ProvinceNum[x] - 1, v = -1;
    int order_kk = 0;
    while (order_kk < jd_ProvinceNum[x])
    {
        order_City[x][order_kk++] = u;
        s[u] = 1;
        for (i = 0; i < jd_ProvinceNum[x] - 1; i++)
        {
            if (u == i)continue;
            if (!s[i] && (v < 0 || zfew[x][u][i] < zfew[x][u][v] || (cost_In_Province[x][u][i] <
cost_In_Province[x][u][v] && zfew[x][u][v]==zfew[x][u][i])))
            {
                v = i;
            }
        }
    }
}

```

```

    }
    u = v;
    v = -1;
}
order_City[x][order_kk++] = jd_ProvinceNum[x] - 1;
double sum = 0;
double hfqqq = 0;
string fname = "E:\\redcp\\研究生工作\\数模\\省内景点\\" + provinceName[x] + ".txt";
FILE * tp;
tp = fopen(fname.c_str(), "w+");
fprintf(tp, "消耗时间(天)                出发地                行车时间(h)
行车里程(公里)                游览景区\n");
for (i = 1; i < order_kk; i++)
    { //出发地、行车时间、行车里程、游览景区
        int u = order_City[x][i - 1], v = order_City[x][i];
        double dis = is90[x][u][v] ? dist_In_Province[x][u][v] * 90 : dist_In_Province[x][u][v] * 40;
        fprintf(tp, "%-20.11f        %-20s        %-20.11f        %-20.11f        %-20s\n",
cost_In_Province[x][u][v],    jd_ProvinceName[x][u].c_str(),    dist_In_Province[x][u][v],    dis,
jd_ProvinceName[x][v].c_str());

        //printf("%d --> %d 花费: %.2lf 天 行车时间: %.2lf 小时 行车公里: %.2lfkm \n",
order_city[i - 1] + 1, order_city[i] + 1, edge[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]] * 90);
        sum += cost_In_Province[x][order_City[x][i - 1]][order_City[x][i]];
        hfqqq += zfew[x][order_City[x][i - 1]][order_City[x][i]];
    }
fclose(tp);
province_Time[x] = sum;
hfp[x] = hfqqq + 300 * (sum - 1); //最后一天不用租车
}

```

```

void load_And_solve_Jd_Info()
{
    int i, j, k;
    for (i = 0; i < N; i++)
    {
        int xxx;
        scanf("%d", &xxx);
        for (j = 0; j < jd_ProvinceNum[i]; j++)
        {
            cin >> jd_Time[i][j];
        }
    }
}

```

```

        cin >> zfm[i][j];
        cin >> zfq[i][j];
    }
    for (j = 0; j < jd_ProvinceNum[i]; j++)
    {
        for (k = 0; k < jd_ProvinceNum[i]; k++)
        {
            scanf("%lf", &dist_In_Province[i][j][k]);
            if (dist_In_Province[i][j][k] <= 40)
            {
                zfew[i][j][k] = dist_In_Province[i][j][k] * 0.6;
                dist_In_Province[i][j][k] /= 40;
                is90[i][j][k] = false;
            }
            else
            {
                zfew[i][j][k] = dist_In_Province[i][j][k];
                dist_In_Province[i][j][k] /= 90;
                is90[i][j][k] = true;
            }
        }
    }
    int xx = 0;
    floyd(i);
    get_Cost(i);
    travle_In_Province(i);
}
int xx = 0;
}

void floyd_dist_Province_To_Province()
{
    for (int k = 0; k < N; k++)
    {
        for (int i = 0; i < N; i++)
        {
            for (int j = 0; j < N; j++)
            {
                if (i == k || j == k)continue;
                if (dist_Province_To_Province[i][j] > dist_Province_To_Province[i][k] +
dist_Province_To_Province[k][j])
                    dist_Province_To_Province[i][j] = dist_Province_To_Province[i][k] +
dist_Province_To_Province[k][j];
            }
        }
    }
}

```

```

        }
    }
}
for (int i = 0; i < N; i++)
{
    for (int j = 0; j < N; j++)
    {
        if (i == j)continue;
        cost_Province_To_Province[i][j] = dist_Province_To_Province[i][j] / (90 * 8) +
province_Time[j];
        dist_Province_To_Province[i][j] /= (90 * 8);
    }
}
}

```

```

void solve_Province_And_Jd_Info()
{
    load_Province_And_Jd_Name_Info();
    load_distOfProvinceInfo();
    load_And_solve_Jd_Info();
    floyd_dist_Province_To_Province();
}

```

```

int order_city[100], order_k;
FILE * tp;
bool xian = false;
int cjcjcjcjc = 0;
bool fcity[N];
void put(int x)
{
    int i;
    if (x == start && xian) return;
    if (x == start)xian = true;
    fprintf(tp, "\n\n");
    fcity[x] = true;
    cjcjcjcjc++;
    for (i = 1; i < jd_ProvinceNum[x]; i++)
    {
        //出发地、行车时间、行车里程、游览景区、消耗时间
        //含每次具体出行方式；每一天的出发地、费用、路途时间、游览景区、每个景区的游览
        时间
        int u = order_City[x][i - 1], v = order_City[x][i];
        double dis = is90[x][u][v] ? dist_In_Province[x][u][v] * 90 : dist_In_Province[x][u][v] * 40;
    }
}

```

```

        double tit = (cost_In_Province[x][u][v] - jd_Time[x][v]) * 8;
        if (tit < 0.01) tit = 0.1 * (rand() % 3 + 1);
        //fprintf(tp,
"%-20.11f      %-20s      %-20.11f      %-20.11f      %-20s\n", cost_In_Province[x][u]
[v], jd_ProvinceName[x][u].c_str(), dist_In_Province[x][u][v], dis, jd_ProvinceName[x][v].c_str());
        fprintf(tp,
"\n%-20.11f      %-20s      %-20s      %-20.11f      %-20.11f      %-20s\n",
cost_In_Province[x][u][v], " 租 车 自 驾 ", jd_ProvinceName[x][u].c_str(), zfew[x][u][v], tit,
jd_ProvinceName[x][v].c_str());
        //printf("%d --> %d 花费: %.21f 天 行车时间: %.21f 小时 行车公里: %.21fkm \n",
order_city[i - 1] + 1, order_city[i] + 1, edge[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]], dist[order_city[i - 1]][order_city[i]] * 90);
        //sum += cost_In_Province[x][order_City[x][i - 1]][order_City[x][i]];
    }
}
double alltime = 0;
void ppt(int u, int v)
{
    double shijian = 0, qian = 0;;
    string way = "";
    if (gts[u][v] < INF)
    {
        shijian = gts[u][v] / (60 * 8);
        way = "高铁";
        qian = gtq[u][v];
    }
    else if (fjq[u][v] < INF)
    {
        way = "飞机";
        qian = fjq[u][v];
        shijian = dist_Province_To_Province[u][v] * 90 / 900;
    }
    else
    {
        shijian = dist_Province_To_Province[u][v];
        way = "坐汽车";
    }
    //fprintf(tp, "消耗时间(天)          出发方式          出发地
费用          路途时间(h)          游览景区\n");
    //
    fprintf(tp,
"\n%-20.11f      %-20s      %-20s      %-20.11f      %-20.11f      %-20s\n",
shijian + jd_Time[u][v], way.c_str(), provinceName[u].c_str(), qian, shijian * 8, provinceName[v].c_str());

```

```

    alltime += hf[u][v];
    put(v);
}
void put2()
{
    string fname = "E:\\23.txt";

    tp = fopen(fname.c_str(), "w+");
    fprintf(tp, "消耗时间(天)                出发方式                出发地
费用(元)                路途时间(h)                游览景区\n");
//                fprintf(tp, "消耗时间(天)                出发地
行车时间(h)                行车里程(公里)                游览景区\n");

    put(order_city[0]);
    int u, v;

    for (int i = 1; i < order_k; i++)
    {
        u = order_city[i - 1], v = order_city[i];
        ppt(u, v);
    }

//以下是有的地方不满足模型没走到，人为刻意刻画去走
for (int i = 0; i < N; i++)
{
    if (!fcity[i])
    {
        fprintf(fp, "\n-----\n");
        ppt(start, i);
    }
}
fprintf(tp, "\n%.2lf    %d\n", alltime, cjejcjejc);

fclose(tp);
}
void travle_Out_Province()
{
    int i;
    bool s[N];
    for (i = 0; i < N; i++)s[i] = 0;

```

```

order_k = 0;
int v_cnt = 1;
year = times = 0;
order_city[order_k++] = start;
while (v_cnt < N)
{
    double time = 0;
    double hfqian = 0;
    int u = start, v = -1;
    times++;
    while (time < 15)
    {
        s[u] = 1;
        for (i = 0; i < N; i++)
        {
            if (u == i || i == start)continue;
            double xxxx = province_Time[start];
            if (!s[i] && (cost_Province_To_Province[u][i] + dist_Province_To_Province[start][i] +
time<=15))
                {
                    if (v < 0 || hf[u][i] < hf[u][v] || (cost_Province_To_Province[u][i]
<cost_Province_To_Province[u][v] && hf[u][v] == hf[u][i]))
                        v = i;
                }
        }
        if (v<0) break;
        time += cost_Province_To_Province[u][v];
        hfqian += hf[u][v];
        u = v;
        order_city[order_k++] = v;
        v_cnt++;
        v = -1;
    }
    if (u == start)break;
    if (time <= 15)order_city[order_k++] = start;
}

//测试使用
double sum = 0, tm = 0;
for (i = 1; i < order_k; i++)
{
    //printf("%d --> %d 花费: %.2lf 天 行车时间: %.2lf 小时 行车公里: %.2lfkm \n",
order_city[i - 1] + 1, order_city[i] + 1, edge[order_city[i - 1]][order_city[i]], dist[order_city[i -
1]][order_city[i]], dist[order_city[i - 1]][order_city[i]] * 90);
}

```

```

    if (order_city[i] != start)
    {
        tm += cost_Province_To_Province[order_city[i - 1]][order_city[i]];
        sum += cost_Province_To_Province[order_city[i - 1]][order_city[i]];
    }
    else
    {
        tm += dist_Province_To_Province[order_city[i - 1]][order_city[i]];
        if (tm >= 30){
            year++; tm = 0; //需要年数
        }
        sum += dist_Province_To_Province[order_city[i - 1]][order_city[i]];
    }
}

}

void solve()
{
    start = 9;
    memset(fcity,0,sizeof(fcity));
    solve_Province_And_Jd_Info();
    travle_Out_Province();
}

int main()
{
    solve();
    put2(); //输出总体方案
    return 0;
}

```