

# 乘公交，看奥运

乌耀益（电子信息工程） 胡东科（软件工程） 单兴（数学与应用数学）

全国一等奖

## 摘要

本文处理的是公交线路选择的问题，需要分别在只考虑公汽线路，考虑公汽线路和地铁线路，和考虑乘交通工具与步行的情况下，解决线路选择问题。针对生活中，没有人会希望换乘太多次。本文把换乘次数限制在了两次的范围内解决问题，在做求解时都给出直达，换乘一次和两次的最优线路选择这三个解，供乘客选择。

本文首先建立了合理的存储结构，采用顺序容器和结构体进行数据构架。顺序容器的这种数据结构有效地避免了用数组保存带来的增减元素和考虑数组大小的麻烦或用稀疏矩阵保存线路关系带来的存储空间浪费。

对于问题一，本文建立了多目标的数学模型，给出了搜索算法，并编程求得了结果。在算法设计时，设计了可以被反复调用的站点直达判断函数，有效地降低了编程的难度，同时也使计算效率得到了提高。

对于问题二，将地铁站和由地铁站联系起来的公汽站点进行等价处理，使得它们之间的换乘关系都能得以描述。然后，同样地建立了数学模型，给出了搜索算法，并编程得到了结果。

对于问题三，定义步行的极限时间，在这个乘客愿意步行的范围内的公汽站点和地铁站进行了处理，使得它们之间也能换乘，只不过换乘时间根据步行时间要重新计算。通过控制步行的极限，可以有效地控制这个范围，极限情况下可以退化为问题二。然后，同样地建立了数学模型，并在原搜索算法的基础上主要通过改变站点直达判断函数给出了新的搜索算法，使搜索算法在考虑步行时也能发挥作用。

在设计搜索算法时，对于直达线路选择采取直接判断，对于换乘一次线路选择采取单向搜索，对于换乘两次线路选择采取双向搜索。其中，核心技术在于站点直达判断函数的复用。

**关键词：**顺序容器 多目标模型 程序调用 布尔函数 双向搜索

## 一、问题重述

我国人民翘首企盼的第 29 届奥运会明年 8 月将在北京举行，届时有大量观众到现场观看奥运比赛，其中大部分人将会乘坐公共交通工具（简称公交，包括公汽、地铁等）出行。这些年来，城市的公交系统有了很大发展，北京市的公交线路已达 800 条以上，使得公众的出行更加通畅、便利，但同时也面临多条线路的选择问题。针对市场需求，某公司准备研制开发一个解决公交线路选择问题的自主查询计算机系统。

为了设计这样一个系统，其核心是线路选择的模型与算法，应该从实际情况出发考虑，满足查询者的各种不同需求。需要解决如下问题：

- 1、仅考虑公汽线路，给出任意两公汽站点之间线路选择问题的一般数学模型与算法。
- 2、同时考虑公汽与地铁线路，解决以上问题。
- 3、假设又知道所有站点之间的步行时间，给出任意两站点之间线路选择问题的数学模型。

## 二、问题分析

本文要处理的问题是公交车乘车最佳路线的选择问题。对于三个问题，分别只考虑公汽，考虑公汽和地铁两种交通工具，最后把步行也考虑进来。

寻找一条最佳路线的问题就是在公交网络各种不同的换乘或直达路线中找到一条最佳的路线。笔者认为，对于乘客而言，最佳是可以多方面的因素的。有的乘客希望能够尽快地到达目的地，有的乘客希望可以尽量地省钱。但有一点，乘客都不会希望换乘的次数太多。例如，为了花的钱尽量地少，可能要换乘好几次，或是十几次才能到达目的地，这显然是不合适的。就换乘次数而言，一般不大于两次。因为换乘对乘客而言是一件相当麻烦的事情，要乘客记住所有换乘的车次，换乘的站点，还要考虑换乘车辆能否很快到达等等，所以综合考虑本文提出了下文中的假设 3。

## 三、模型假设

- 1) 公汽、地铁的行驶都是匀速的，同类交通工具相邻站点的间距是相同的。
- 2) 上下公汽、地铁的时间间隙忽略不计。
- 3) 不考虑需要换乘 2 次以上的乘车路线，除非在换乘 2 次及以下的路线中不存在能到达目的地的乘车路线。

## 四、变量声明

$s_i$ : 站点（包括公共汽车站点和地铁站点）；  
 $L_i$ : 交通工具（包括公共汽车和地铁）；  
 $t(s_i, s_j)$ : 站点  $s_i$  到  $s_j$  所需花费的时间（分钟）；  
 $w(s_i)$ : 在站点  $s_i$  换乘所需花费的时间（分钟）；

$g(s_i, s_j)$ : 乘同一交通工具从站点  $s_i$  直达站点  $s_j$  经过的站点数 (包括  $s_j$ );

$f(s_i, s_j)$ : 乘同一交通工具从站点  $s_i$  直达站点  $s_j$  所需的费用 (元);

$C$ : 给定任意两个站点后的以它们为起讫点的路线的集合;

$c_i$ : 给定任意两个站点后的以它们为起讫点的一条路线;

$T(c_i)$ : 路线  $c_i$  上所花费的时间;

$F(c_i)$ : 路线  $c_i$  上所需费用;

## 五、数据构架

### 5.1 数据预处理

为了在程序设计和求解过程中, 尽可能地降低时间复杂度, 使想得到的结果尽快地呈现在使用该系统的用户面前, 我们在将数据输入到程序前, 首先对原始数据进行预处理, 让它适合程序的输入要求。

我们采用 Editplus 2 软件进行替换操作, 可非常方便地得到我们想要的数

据, 处理过程如下:

- 1、字母标记“R”替换汉字“分段计价”;
- 2、字母标记“Y”替换汉字“单一票制 1 元”;
- 3、字母标记“F”替换汉字“上行”;
- 4、字母标记“B”替换汉字“下行”;
- 5、字母标记“C”替换汉字“环形”;
- 6、结束标记“E”替换结束符“END”, 无“END”则手动加入结束标记“E”;
- 7、去掉符号“。”、“:”、“—”和地铁数据中的“票价 3 元, 本线路使用, 并可换乘 T2”, 如果去掉后使前后数据连在一起则在它们之间插入空格;
- 8、原路返回的路线, 在始发站的数据前插入字母标记“X”;
- 9、将字母+数字的数据中间插入一个空格变为字母+空格+数字;

举例一:

L001

分段计价。

S0619-S1914-S0388-S0348-S0392-S0429-S0436-S3885-S3612-S0819-S3524-S0820-S3914-S0128-S0710

处理后:

L 001

R

X S 0619 S 1914 S 0388 S 0348 S 0392 S 0429 S 0436 S 3885 S 3612 S 0819 S 3524 S 0820 S 3914 S 0128 S 0710

举例二:

L425

单一票制 1 元。

环行: S1042-S0130-S3019-S0969-S3741-S1963-S3741-S3019-S0477-S1042

处理后:

L 425

Y

C S 1042 S 0130 S 3019 S 0969 S 3741 S 1963 S 3741 S 3019 S 0477 S 1042

举例三:

L520

单一票制 1 元。

上行: S0600-S2861-S0601-S2365-S2634-S1284-S1644-S1643-S2028-S3221-S1344-S2645-S1848

```

下行：S1848-S2645-S1344-S3221-S2028-S1643-S1644-S1284-S2634-
S2365-S2796-S2861-S2027-S0600
END
处理后：
L 520
Y
F S 0600 S 2861 S 0601 S 2365 S 2634 S 1284 S 1644 S 1643 S 2028 S 3221
S 1344 S 2645 S 1848
B S 1848 S 2645 S 1344 S 3221 S 2028 S 1643 S 1644 S 1284 S 2634 S 2365
S 2796 S 2861 S 2027 S 0600
E

```

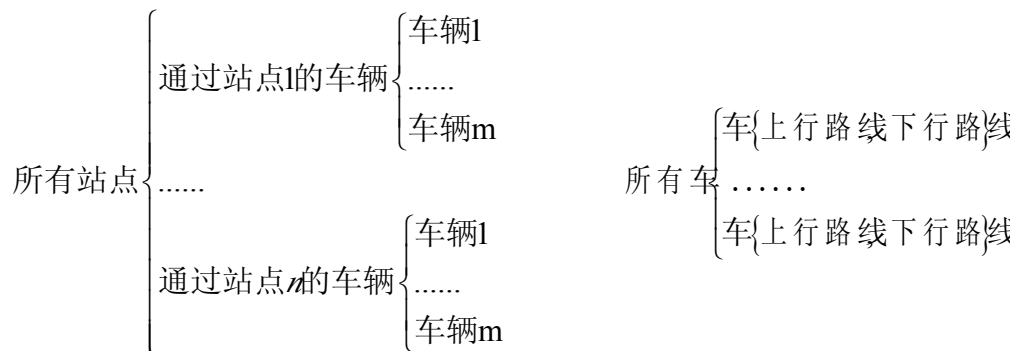
## 5.2 存储结构

在进行程序设计的过程前，事先构架一个良好的存储结构，无疑对整个程序设计起着至关重要的作用，特别在这道大数据量的问题前更显得尤为重要，所以对于如何构架进行了多次尝试后，得到了一个理想的存储结构。

本文采用结构体和 **Vector** 顺序容器相结合的数据结构。**Vector** 顺序容器类似于栈和一维数组，但功能比栈和数组都强大得多，不仅是自增长的，无须知道输入数据的多少，可以通过自身的 **push\_back()** 函数和 **pop\_back()** 函数进行无限量的插入删除操作，而且通过调用 **size()** 函数很方便地知道它存储的数据的多少，无须用数字单独记录特别适合于本题的数据结构设计，这样我们无须手动计算各个路线的长度，而且各个路线的长度是不相同的。

根据本题的数据，我们可以得知此数据不仅属于大数据量，而且路线的长度不同，差别比较大，若用数组来存储的话，无疑数组开销非常大，造成严重的空间浪费，而且还需手动统计线路长度，而这些问题用 **Vector** 顺序容器就能得到很好的解决。

数据组织用文字表述如下：



将上述数据组织转换为代码为：

问题一存储结构定义	问题二存储结构定义
<pre>typedef struct {     char F_B;     int i;     int num; }car;  vector&lt; vector&lt;car&gt; &gt; station(3958);  struct node {     vector&lt;int&gt; forward;     vector&lt;int&gt; backward;     char R_Y;     char X_FB_C; }Car[521];</pre>	<pre>typedef struct {     char F_B;     int i;     int num;     char C_R; }tool;  vector&lt; vector&lt;tool&gt; &gt; station(4000);  struct node {     vector&lt;int&gt; forward;     vector&lt;int&gt; backward;     char R_Y;     char X_FB_C;     char C_R; }Tool[523];  vector&lt; vector&lt;int&gt; &gt; Link(40);</pre>

由于问题二在问题一的基础上考虑了地铁线路，比问题一的数据构架定义更具有代表性，所以我们以问题二的数据结构定义为例来进行文字说明。

结构体tool定义了程序设计中经过站点时要用到的交通工具的部分属性集，其属性有交通工具的编号num，交通工具在站点上的序号i（程序中我们按交通工具的路线先后顺序进行了标号），属于哪种交通工具的标记C\_R,属于上行路线或下行路线的标记F\_B;

容器station定义了所有站点的集合，其属性vector<tool>表示经过某个站点的交通工具tool的总和；

结构体Tool定义了一个完整的交通工具的属性集，其属性有上行路线forward，下行路线backward，属于哪种交通工具的标记C\_R,计价方式的标记R\_Y,行车方式的标记X\_FB\_C（其中环形的行车方式归于上行路线，下行路线为空，原路返回的上行路线和下行路线正好相反）；

容器Link定义了与每个地铁站点邻接的汽车站点的集合，其属性vector<int>就是与某个地铁站点邻接的汽车站点编号的集合；

数据结构的关系直观地表达如下：

$$\begin{aligned}
& \left\{ \begin{array}{l} station \left\{ \begin{array}{l} tool\{num,i,C\_R,F\_B\} \\ \dots\dots \\ tool\{num,i,C\_R,F\_B\} \end{array} \right. \\ \dots\dots \\ station \left\{ \begin{array}{l} tool\{num,i,C\_R,F\_B\} \\ \dots\dots \\ tool\{num,i,C\_R,F\_B\} \end{array} \right. \end{array} \right. \\
AllStation & \left\{ \begin{array}{l} Tool\{forward\{station_1,\dots,station_n\},backward\{station_1,\dots,station_m\},C\_R,F\_B,X\_FB\_C\} \\ \dots\dots \\ Tool\{forward\{station_1,\dots,station_n\},backward\{station_1,\dots,station_m\},C\_R,F\_B,X\_FB\_C\} \end{array} \right. \\
AllTool &
\end{aligned}$$

## 六、问题求解

### 6.1 问题一

#### 6.1.1 数学模型

对于任意两个站点  $s_0$  和  $s_n$ ，设以  $s_0$  为起始点， $s_n$  为目的地的路线有  $m$  条，路线集合  $C = \{c_1, c_2, \dots, c_m\}$ ，其中任意一条路线  $c_i = (s_0, s_1, s_2, \dots, s_{n-1}, s_n)$ ，路线  $c_i$  中有  $q$  个站点换乘，分别为  $s_{j1}, s_{j2}, \dots, s_{jq}$ ，由于只考虑公汽线路，所以任一路线  $c_i$  上的站点均为公共汽车站，则由题意可得  $f(s_i, s_j)$  与  $g(s_i, s_j)$  的关系如下：

$$f(s_i, s_j) = \begin{cases} 1, & 1 \leq g(s_i, s_j) \leq 20 \\ 2, & 21 \leq g(s_i, s_j) \leq 40 \\ 3, & g(s_i, s_j) \geq 41 \\ 1, & \text{若汽车单一计价} \end{cases}$$

且  $t(s_k, s_{k+1}) = 3$ ， $u(s_i) = 5$ ，则路线  $c_i$  上所需费用

$$F(c_i) = \begin{cases} f(s_0, s_n), & q = 0 \\ f(s_0, s_{j1}) + f(s_{j1}, s_n), & q = 1 \\ f(s_0, s_{j1}) + \sum_{k=1}^{q-1} f(s_{jk}, s_{j(k+1)}) + f(s_{jq}, s_n), & q \geq 2 \end{cases}$$

令决策变量  $x_i = \begin{cases} 1, & \text{在站点 } s_i \text{ 换乘} \\ 0, & \text{否则} \end{cases}$ ，则  $q = \sum_{i=1}^{n-1} x_i$

路线  $c_i$  上所花费的时间  $T(c_i) = \sum_{k=0}^{n-1} t(s_k, s_{k+1}) + \sum_{i=1}^{n-1} x_i u(s_i)$

设  $A = \{L_1, L_2, \dots, L_h\}$  表示经过起始点  $s_0$  的所有交通工具的集合， $B = \{L'_1, L'_2, \dots, L'_u\}$  表示到达目的地  $s_n$  的所有交通工具的集合，

令  $y_i = \begin{cases} 1, & \text{在起始点 } s_0 \text{ 处选择交通工具 } L_i \\ 0, & \text{否则} \end{cases}$ ， $z_i = \begin{cases} 1, & \text{选择交通工具 } L'_i \text{ 到达目的地 } s_n \\ 0, & \text{否则} \end{cases}$

则分别以花费时间最小和所需费用最小为目标，建立多目标规划模型：

$$\begin{aligned} & \min T(c_i) \\ & \min F(c_i) \quad , \quad \forall c_i \in C \\ & s.t. \quad \begin{cases} \sum_{i=1}^{n-1} x_i \leq 2 \\ \sum_{i=1}^h y_i = 1 \\ \sum_{i=1}^u z_i = 1 \end{cases} \end{aligned}$$

### 6.1.2 算法设计思路

由于我们采用了良好的数据构架（在第五部分已详细指出），在算法设计中我们根据问题一要求只考虑公汽线路的情况下，建立了以费用和时间为主次目标的两个模型来分别求解，但其算法的基本思想是完全相同的，在选择其他目标来求解，也完全可以用这个算法设计思路来求解，也就体现了它具有很好的通用性。

由于所要取得的直达路线，换乘一次的路线，换乘二次的路线都是基于对判断有无直达线路的布尔型函数  $Fuction(X, Y)$  的调用，所以函数  $Fuction(X, Y)$  也就成了算法中关键的部分，在下面的算法设计思路中也多次用到，这里给出它的定义为：如果经过起始站点  $X$  和目的站点  $Y$  有相同车辆，且车辆在两个站点的行车方向相同，目的站点  $Y$  的行车序号要大于起始站点  $X$  的行车序号，就认为  $X$  和  $Y$  直达。

基于存储结构的伪代码  $Fuction(X, Y)$  表述如下：

```
bool Fuction(X, Y){
    for(i=0; i<up.size(); i++)
        for(j=0; j<down.size(); j++)
            if(X[i].num == Y[j].num && X[i].F_B == Y[j].F_B && X[i].i < Y[j].i)
                return true;
    return false;
}
```

我们首先将算法设计思路列出来，再在下面的两个模型中进行求解。

算法设计思路描述：

**Step0:** 从文件中读取数据并保存在存储结构中；

**Step1:** 输入起始站点  $s_1$  和目标站点  $s_2$ ；

**Step2:** 对于起始站点  $s_1$  与目标站点  $s_2$  首先调用子函数  $Fuction(s_1, s_2)$  判断是否可以直达，若返回真则进行目标选择求得最优解，并将结果输出；

**Step3:** 从起始站点  $s_1$  向前搜索每一条通过它的路线上的站点  $s_i$ （如图 1），并调用子函数  $Fuction(s_i, s_2)$  判断是否可以直达，若返回真则可以通过换乘一次到达目的地，并进行目标选择求得最优解，并将结果输出；

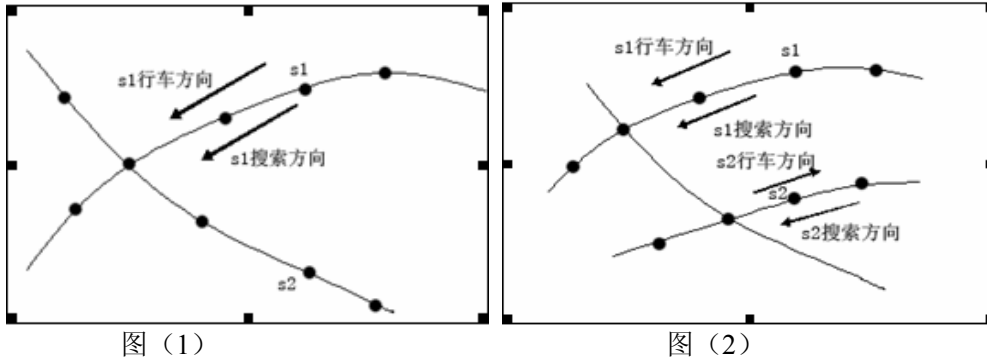


图 (1)

图 (2)

Step4: 从起始站点  $s_1$  向前搜索每一条通过它的路线上的站点  $s_i$ , 再从目标站点  $s_2$  向后搜索每一条通过它的路线上的站点  $s_j$  (如图 2), 调用子函数  $Fuction(s_i, s_j)$  判断是否可以直达, 若返回真则可以通过换乘二次到达目的地, 并进行目标选择求得最优解, 并将结果输出;

这里我们输出直达, 换乘一次, 换乘二次三个结果, 目的在于给使用系统的用户提供更多的选择余地。

### 6.1.3 费用优先, 时间次优的模型

基于算法设计思路, 我们以费用优先, 时间次优作为目标来进行求解, 每个步骤的目标选择为 (符号可参见变量说明):

Step2 中的目标选择为: 根据费用  $f(s_1, s_2)$  最小原则进行选择。如果选择过程中, 费用  $f(s_1, s_2)$  相等则按时间  $t(s_1, s_2)$  最小原则进行选择,

Step3 中的目标选择为: 根据费用  $f(s_1, s_2) = f(s_1, s_i) + f(s_i, s_2)$  最小原则进行选择。如果选择过程中, 费用  $f(s_1, s_2)$  相等则按时间  $t(s_1, s_2) = t(s_1, s_i) + t(s_i, s_2) + u(s_i)$  最小原则进行选择;

Step4 中的目标选择为: 根据费用  $f(s_1, s_2) = f(s_1, s_i) + f(s_i, s_j) + f(s_j, s_2)$  最小原则进行选择。如果选择过程中, 费用  $f(s_1, s_2)$  相等则按时间  $t(s_1, s_2) = t(s_1, s_i) + t(s_i, s_j) + t(s_j, s_2) + u(s_i) + u(s_j)$  最小原则进行选择;

根据上面所给的目标选择设计程序 (参见附录 1), 运行一次可得到直达, 换乘一次和换乘二次的结果为:

#### **(1) S3359 → S1828**

从 S3359 到 S1828 无直达车

从 S3359 到 S1828 可以先乘坐 L436 在 S1784 换乘 L167

所需费用为:3 所花时间为:101

L436 行车路线为:S3359->S2026->S1132->S2266->S2263->S3917->S2303->S2301->S3233->S618->S616->S2112->S2110->S2153->S2814->S2813->S3501->S3515->S3500->S756->S492->S903->S1768->S955->S480->S2703->S2800->S2192->S2191->S1829->S3649->S1784

L167 行车路线为:S1784->S1828

从 S3359 到 S1828 可以先乘坐 L15 在 S2903 换乘 L201 后再在 S1671 换乘 L41

所需费用为:3 所花时间为:73

L15 行车路线为:S3359->S2903



L201 行车路线为:S2903->S2027->S1327->S1842->S609->S483->S604->S2650->S3470->S2619->S2340->S2182->S992->S2322->S1770->S1790->S458->S1792->S1783->S1671

L41 行车路线为:S1671->S1828

---

### **(2) S1557 → S481**

从 S1557 到 S481 无直达车

从 S1557 到 S481 不能通过换乘一次车来达到目的地

从 S1557 到 S481 可以先乘坐 L84 在 S1919 换乘 L189 后再在 S3186 换乘 L460  
所需费用为:3 所花时间为:106

L84 行车路线为:S1557->S3158->S2628->S3408->S2044->S1985->S2563->S2682->S28->S29->S55->S51->S1919

L189 行车路线为:S1919->S2840->S1402->S3186

L460 行车路线为:S3186->S3544->S2116->S2119->S1788->S1789->S1770->S2322->S992->S2184->S2954->S3117->S2424->S1174->S902->S903->S2101->S481

---

### **(3) S971 → S485**

从 S971 到 S485 无直达车

从 S971 到 S485 可以先乘坐 L13 在 S2184 换乘 L417

所需费用为:3 所花时间为:128

L13 行车路线为:S971->S3832->S3341->S2237->S3565->S3333->S1180->S3494->S1523->S1520->S1988->S1743->S1742->S1181->S1879->S3405->S2517->S3117->S2954->S531->S2184

L417 行车路线为:S2184->S992->S2322->S1770->S1789->S2119->S2116->S3544->S3186->S3409->S2717->S1402->S2840->S643->S2079->S1920->S2480->S2482->S2210->S3332->S3351->S485

从 S971 到 S485 可以先乘坐 L13 在 S1609 换乘 L140 后再在 S2654 换乘 L469

所需费用为:3 所花时间为:106

L13 行车路线为:S971->S3571->S1609

L140 行车路线为:S1609->S3242->S1481->S3426->S2553->S3903->S1553->S3531->S1967->S12->S2636->S2113->S2112->S2833->S618->S1327->S2303->S2263->S3037->S2654

L469 行车路线为:S2654->S1729->S3766->S1691->S1383->S1381->S1321->S2019->S2017->S2159->S772->S485

---

### **(4) S8 → S73**

从 S8 到 S73 无直达车

从 S8 到 S73 可以先乘坐 L159 在 S400 换乘 L474

所需费用为:2 所花时间为:83

L159 行车路线为:S8->S3412->S2743->S3586->S2544->S913->S2953->S3874->

S630->S854->S400

L474 行车路线为:S400->S2633->S3053->S410->S411->S2846->S605->S604->S527->S525->S3470->S2619->S2340->S3162->S2181->S2705->S73

从 S8 到 S73 可以先乘坐 L198 在 S3766 换乘 L296 后再在 S2184 换乘 L345  
所需费用为:3 所花时间为:67

L198 行车路线为:S8->S1383->S1691->S3766

L296 行车路线为:S3766->S1729->S2654->S3231->S3917->S2303->S1327->S618->S3100->S2151->S3746->S3501->S2517->S2184

L345 行车路线为:S2184->S3162->S2181->S73

---

#### **(5) S148 → S485**

从 S148 到 S485 无直达车

从 S148 到 S485 不能通过换乘一次车来达到目的地

从 S148 到 S485 可以先乘坐 L308 在 S36 换乘 L156 后再在 S3351 换乘 L417  
所需费用为:3 所花时间为:106

L308 行车路线为:S148->S462->S361->S1797->S2221->S302->S2222->S2737->S1716->S128->S2268->S1308->S1391->S2272->S36

L156 行车路线为:S36->S3233->S618->S617->S721->S2057->S2361->S608->S399->S2535->S2534->S239->S497->S2090->S2082->S2210->S3332->S3351

L417 行车路线为:S3351->S485

---

#### **(6) S87 → S3676**

从 S87 到 S3676 无直达车

从 S87 到 S3676 可以先乘坐 L454 在 S3496 换乘 L209

所需费用为:2 所花时间为:65

L454 行车路线为:S87->S857->S630->S1427->S1426->S541->S978->S3389->S1919->S641->S2840->S3496

L209 行车路线为:S3496->S1883->S1159->S2699->S2922->S3010->S583->S1987->S82->S3676

从 S87 到 S3676 可以先乘坐 L21 在 S88 换乘 L231 后再在 S427 换乘 L97

所需费用为:3 所花时间为:46

L21 行车路线为:S87->S88

L231 行车路线为:S88->S609->S483->S604->S2650->S3693->S1659->S2962->S622->S456->S427

L97 行车路线为:S427->S3676

---

### **6.1.4 费用优先，时间次优的模型**

基于算法设计思路，我们以时间优先，费用次优作为目标来进行求解，每个步骤的目标选择为（符号可参见变量说明）：

Step2 中的目标选择为：根据时间  $t(s_1, s_2)$  最小原则进行选择。如果选择过程中，费

用  $\alpha(s_1, s_2)$  相等则按时间  $f(s_1, s_2)$  最小原则进行选择,

Step3 中的目标选择为: 根据费用  $\alpha(s_1, s_2) = \alpha(s_1, s_i) + \alpha(s_i, s_2) + w(s_i)$  最小原则进行选择。如果选择过程中, 费用  $\alpha(s_1, s_2)$  相等则按时间  $f(s_1, s_2) = f(s_1, s_i) + f(s_i, s_2)$  最小原则进行选择;

Step4 中的目标选择为: 根据费用  $\alpha(s_1, s_2) = \alpha(s_1, s_i) + \alpha(s_i, s_j) + \alpha(s_j, s_2) + w(s_i) + w(s_j)$  最小原则进行选择。如果选择过程中, 费用  $f(s_1, s_2)$  相等则按时间  $f(s_1, s_2) = f(s_1, s_i) + f(s_i, s_j) + f(s_j, s_2)$  最小原则进行选择;

根据上面所给的目标选择设计程序(参见附录2), 运行一次可得到直达, 换乘一次和换乘二次的结果为:

### **(1) S3359 → S1828**

从 S3359 到 S1828 无直达车

从 S3359 到 S1828 可以先乘坐 L436 在 S1784 换乘 L167

所花时间为:101 所需费用为:3

L436 行车路线为:S3359->S2026->S1132->S2266->S2263->S3917->S2303->  
S2301->S3233->S618->S616->S2112->S2110->S2153->S2814->S2813->  
S3501->S3515->S3500->S756->S492->S903->S1768->S955->S480->S2703->  
S2800->S2192->S2191->S1829->S3649->S1784

L167 行车路线为:S1784->S1828

从 S3359 到 S1828 可以先乘坐 L15 在 S2903 换乘 L201 后再在 S1671 换乘 L41

所花时间为:73 所需费用为:3

L15 行车路线为:S3359->S2903

L201 行车路线为:S2903->S2027->S1327->S1842->S609->S483->S604->S2650->  
S3470->S2619->S2340->S2182->S992->S2322->S1770->S1790->S458->S1792->  
S1783->S1671

L41 行车路线为:S1671->S1828

---

### **(2) S1557 → S481**

从 S1557 到 S481 无直达车

从 S1557 到 S481 不能通过换乘一次车来达到目的地

从 S1557 到 S481 可以先乘坐 L84 在 S1919 换乘 L189 后再在 S3186 换乘 L460

所花时间为:106 所需费用为:3

L84 行车路线为:S1557->S3158->S2628->S3408->S2044->S1985->S2563->  
S2682->S28->S29->S55->S51->S1919

L189 行车路线为:S1919->S2840->S1402->S3186

L460 行车路线为:S3186->S3544->S2116->S2119->S1788->S1789->S1770->  
S2322->S992->S2184->S2954->S3117->S2424->S1174->S902->S903->  
S2101->S481

---

### **(3) S971 → S485**

从 S971 到 S485 无直达车

从 S971 到 S485 可以先乘坐 L13 在 S2184 换乘 L417

所花时间为:128 所需费用为:3

L13 行车路线为:S971->S3832->S3341->S2237->S3565->S3333->S1180->  
S3494->S1523->S1520->S1988->S1743->S1742->S1181->S1879->S3405->  
S2517->S3117->S2954->S531->S2184

L417 行车路线为:S2184->S992->S2322->S1770->S1789->S2119->S2116->S3544->  
S3186->S3409->S2717->S1402->S2840->S643->S2079->S1920->S2480->S2482->  
S2210->S3332->S3351->S485

从 S971 到 S485 可以先乘坐 L13 在 S1609 换乘 L140 后再在 S2654 换乘 L469

所花时间为:106 所需费用为:3

L13 行车路线为:S971->S3571->S1609

L140 行车路线为:S1609->S3242->S1481->S3426->S2553->S3903->S1553->  
S3531->S1967->S12->S2636->S2113->S2112->S2833->S618->S1327->S2303->  
S2263->S3037->S2654

L469 行车路线为:S2654->S1729->S3766->S1691->S1383->S1381->S1321->  
S2019->S2017->S2159->S772->S485

---

#### **(4) S8 → S73**

从 S8 到 S73 无直达车

从 S8 到 S73 可以先乘坐 L159 在 S400 换乘 L474

所花时间为:83 所需费用为:2

L159 行车路线为:S8->S3412->S2743->S3586->S2544->S913->S2953->S3874->  
S630->S854->S400

L474 行车路线为:S400->S2633->S3053->S410->S411->S2846->S605->S604->  
S527->S525->S3470->S2619->S2340->S3162->S2181->S2705->S73

从 S8 到 S73 可以先乘坐 L198 在 S3766 换乘 L296 后再在 S2184 换乘 L345

所花时间为:67 所需费用为:3

L198 行车路线为:S8->S1383->S1691->S3766

L296 行车路线为:S3766->S1729->S2654->S3231->S3917->S2303->S1327->  
S618->S3100->S2151->S3746->S3501->S2517->S2184

L345 行车路线为:S2184->S3162->S2181->S73

---

#### **(5) S148 → S485**

从 S148 到 S485 无直达车

从 S148 到 S485 不能通过换乘一次车来达到目的地

从 S148 到 S485 可以先乘坐 L308 在 S36 换乘 L156 后再在 S3351 换乘 L417

所花时间为:106 所需费用为:3

L308 行车路线为:S148->S462->S361->S1797->S2221->S302->S2222->S2737->  
S1716->S128->S2268->S1308->S1391->S2272->S36

L156 行车路线为:S36->S3233->S618->S617->S721->S2057->S2361->S608->

S399->S2535->S2534->S239->S497->S2090->S2082->S2210->S3332->S3351  
L417 行车路线为:S3351->S485

---

### **(6) S87 → S3676**

从 S87 到 S3676 无直达车

从 S87 到 S3676 可以先乘坐 L454 在 S3496 换乘 L209  
所花时间为:65 所需费用为:2

L454 行车路线为:S87->S857->S630->S1427->S1426->S541->S978->S3389->  
S1919->S641->S2840->S3496

L209 行车路线为:S3496->S1883->S1159->S2699->S2922->S3010->S583->  
S1987->S82->S3676

从 S87 到 S3676 可以先乘坐 L21 在 S88 换乘 L231 后再在 S427 换乘 L97  
所花时间为:46 所需费用为:3

L21 行车路线为:S87->S88

L231 行车路线为:S88->S609->S483->S604->S2650->S3693->S1659->S2962->  
S622->S456->S427

L97 行车路线为:S427->S367

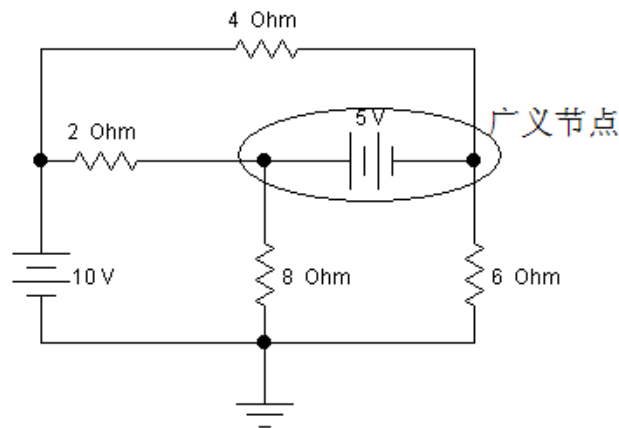
---

## **6.2 问题二**

### **6.2.1 广义站点概念引入**

对于问题二，既要考虑公汽线路，也要考虑地铁线路。那么，考虑的站点也包括了地铁站和公汽车站两种。而地铁站和汽车站之间又存在着联系：同一地铁站对应的任意两个公汽车站之间可以通过地铁站换乘(无需支付地铁费)。

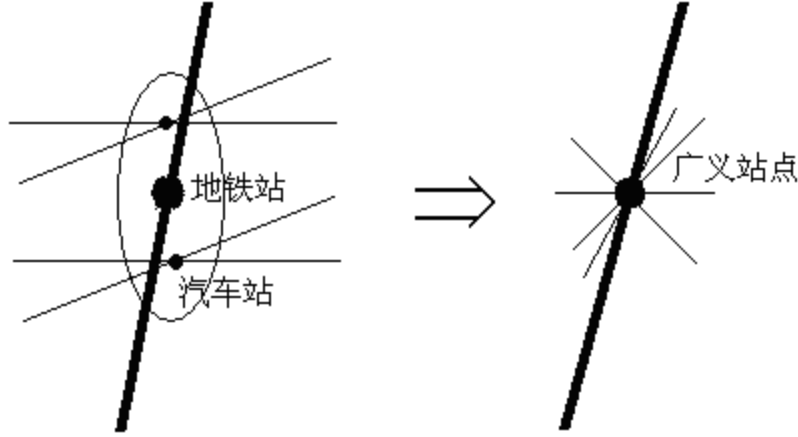
所以我们联想到电路分析中的广义节点的概念与思想，提出广义站点的概念，完全了汽车站和地铁站的换乘关系。如下图是个有广义节点的电路<sup>[2]</sup>：



图中的黑点就是电路中电路的节点，常用于电路分析的节点分析。而广义节点可视为包含电路原器件的一个闭合区。根据广义节点列出的基尔霍夫电流定律的方程和一般节点在形式上是统一的。基尔霍夫电流定律是指流入节点的电流和流出节点的电流相等。用数学表达式表示为： $\sum I_{in} = \sum I_{out}$ 。这种形式上的统一，为我们分析问题提供了有利武器，同时，在使用上也很方便。

回到公交线路选择的问题上来，我们也可以相似地定义广义站点。把因地铁站联系

起来的汽车站和地铁站用广义站点表示出来。把包括且仅包括这些站点的范围用广义站点来表示，甚至可以抽象成一个具体的站点，所有进出这个范围的公汽和地铁都可以相互换乘，就像是在一个站点一样。这样，与之平行的，也可认为所有可以因这个地铁站换乘的汽车站和这个地铁站，这些站点都可以看作等价，能换乘通过这个广义站点里的任何现实站点的所有线路。如下图所示：



对于广义站点，有四种不同的换乘方式，所以广义站点的换乘时间  $w$  不是如实际站点一样是一个定值，而有四种取值，视进出站状况而定，具体如下：

$$w = \begin{cases} 5, & \text{进站和出站都是公汽} \\ 4, & \text{进站和出站都是地铁} \\ 7, & \text{进站是地铁，出站是公汽} \\ 6, & \text{进站是公汽，出站是地铁} \end{cases}$$

这样处理就能同等对待那些本来不能通过公汽换乘但能通过地铁换乘的站点了。

### 6.2.2 数学模型

根据上述处理，可以写出问题二的数学模型如下：

对于任意两个站点  $s_0$  和  $s_n$ （这里的站点和以下的站点都可以是广义站点），设以  $s_0$  为起始点， $s_n$  为目的地的路线有  $m$  条，路线集合  $C = \{c_1, c_2, \dots, c_m\}$ ，其中任意一条路线  $c_i = (s_0, s_1, s_2, \dots, s_{n-1}, s_n)$ ，路线  $c_i$  中有  $q$  个站点换乘，分别为  $s_{j1}, s_{j2}, \dots, s_{jq}$ ，由于同时考虑公汽与地铁线路，所以任一路线  $c_i$  上的站点为公共汽车站或地铁站，则  $f(s_i, s_j)$  与

$$g(s_i, s_j) \text{ 的关系如下: } f(s_i, s_j) = \begin{cases} 3, & \text{若乘地铁} \\ 1, & \text{若乘汽车单一计价} \\ 1, & \text{若乘汽车, 且 } 1 \leq g(s_i, s_j) \leq 20 \\ 2, & \text{若乘汽车, 且 } 21 \leq g(s_i, s_j) \leq 40 \\ 3, & \text{若乘汽车, 且 } g(s_i, s_j) \geq 41 \end{cases}$$

$$\text{且 } t(s_k, s_{k+1}) = \begin{cases} 3, & \text{如果在相邻站点 } s_k \text{ 与 } s_{k+1} \text{ 通过乘汽车到达} \\ 2.5, & \text{如果在相邻站点 } s_k \text{ 与 } s_{k+1} \text{ 通过乘地铁到达} \end{cases}$$

$$w(s_i) = \begin{cases} 5, & \text{在站点 } s_i \text{ 处公汽换乘公汽} \\ 4, & \text{在站点 } s_i \text{ 处地铁换乘地铁} \\ 7, & \text{在站点 } s_i \text{ 处地铁换乘公汽} \\ 6, & \text{在站点 } s_i \text{ 处公汽换乘地铁} \end{cases}$$

则路线  $c_i$  上所需费用

$$F(c_i) = \begin{cases} f(s_0, s_n), & q = 0 \\ f(s_0, s_{j1}) + f(s_{j1}, s_n), & q = 1 \\ f(s_0, s_{j1}) + \sum_{k=1}^{q-1} f(s_{jk}, s_{j(k+1)}) + f(s_{jq}, s_n), & q \geq 2 \end{cases}$$

$$\text{令决策变量 } x_i = \begin{cases} 1, & \text{在站点 } s_i \text{ 换乘} \\ 0, & \text{否则} \end{cases}, \text{ 则 } q = \sum_{i=1}^{n-1} x_i$$

$$\text{路线 } c_i \text{ 上所花费的时间 } T(c_i) = \sum_{k=0}^{n-1} t(s_k, s_{k+1}) + \sum_{i=1}^{n-1} x_i w(s_i)$$

设  $A = \{L_1, L_2, \dots, L_h\}$  表示经过起始点  $s_0$  的所有交通工具的集合,  $B = \{L'_1, L'_2, \dots, L'_u\}$  表示到达目的地  $s_n$  的所有交通工具的集合,

$$\text{令 } y_i = \begin{cases} 1, & \text{在起始点 } s_0 \text{ 处选择交通工具 } L_i \\ 0, & \text{否则} \end{cases}$$

$$z_i = \begin{cases} 1, & \text{选择交通工具 } L'_i \text{ 到达目的地 } s_n \\ 0, & \text{否则} \end{cases}$$

则分别以花费时间最小和所需费用最小为目标, 建立多目标规划模型:

$$\begin{aligned} \min & \quad T(c_i) \\ \min & \quad F(c_i) \end{aligned}, \quad \forall c_i \in C$$

$$s.t. \quad \begin{cases} \sum_{i=1}^{n-1} x_i \leq 2 \\ \sum_{i=1}^h y_i = 1 \\ \sum_{i=1}^u z_i = 1 \end{cases}$$

### 6.2.3 算法设计思路

在算法设计中, 我们根据问题二要求在公汽线路的基础上加入了地铁线路和地铁站点及公汽站点的相连关系的情况下, 建立了以费用和时间为主次目标的两个模型来分别求解, 但其算法的基本思想也是完全相同的, 在选择其他目标来求解时也完全可以用这个算法设计思路, 也就体现了它具有很好的通用性。

我们将地铁站点及线路和公汽站点及线路统一称为站点和线路, 地铁站点与与其相连的公汽站点经过的所有车和地铁都可以连接所有的这些站点, 这样就可以在问题一的算法设计思路的基础上来设计问题二的算法设计思路, 其中我们可以对地铁站点及线路和公汽站点和线路采用标记法来区分。

针对各别数据,我们增加假设为:当起始站点和目的站点是地铁站与其相连的公汽站点中的其中一个时,忽略在这些站点间的平均耗时。

这里我们首先将算法设计思路列出来,再在下面的两个模型中进行求解。

算法设计思路描述:

Step0: 从文件中读取数据并保存在存储结构中;

Step1: 输入起始站点  $s_1$  和目标站点  $s_2$ ;

Step2: 对于起始站点  $s_1$  与目标站点  $s_2$  首先调用子函数  $Fuction(s_1, s_2)$  判断是否可以直达,若返回真则进行目标选择求得最优解,并将结果输出;

Step3: 从起始站点  $s_1$  向前搜索每一条通过它的路线上的站点  $s_i$ , 并调用子函数  $Fuction(s_i, s_2)$  判断是否可以直达,若返回真则可以通过换乘一次到达目的地,并进行目标选择求得最优解,并将结果输出;

Step4: 从起始站点  $s_1$  向前搜索每一条通过它的路线上的站点  $s_i$ , 再从目标站点  $s_2$  向后搜索每一条通过它的路线上的站点  $s_j$ , 调用子函数  $Fuction(s_i, s_j)$  判断是否可以直达,若返回真则可以通过换乘二次到达目的地,并进行目标选择求得最优解,并将结果输出;

这里我们也输出直达,换乘一次,换乘二次三个结果,目的在于给使用系统的用户提供更多的选择余地。

#### 6.2.4 费用优先,时间次优的模型

基于算法设计思路,我们以费用优先,时间次优作为目标来进行求解,每个步骤的目标选择为(符号可参见变量说明):

Step2 中的目标选择为:根据费用  $f(s_1, s_2)$  最小原则进行选择。如果选择过程中,费用  $f(s_1, s_2)$  相等则按时间  $t(s_1, s_2)$  最小原则进行选择,

Step3 中的目标选择为:根据费用  $f(s_1, s_2) = f(s_1, s_i) + f(s_i, s_2)$  最小原则进行选择。如果选择过程中,费用  $f(s_1, s_2)$  相等则按时间  $t(s_1, s_2) = t(s_1, s_i) + t(s_i, s_2) + w(s_i)$  最小原则进行选择;

Step4 中的目标选择为:根据费用  $f(s_1, s_2) = f(s_1, s_i) + f(s_i, s_j) + f(s_j, s_2)$  最小原则进行选择。如果选择过程中,费用  $f(s_1, s_2)$  相等则按时间  $t(s_1, s_2) = t(s_1, s_i) + t(s_i, s_j) + t(s_j, s_2) + w(s_i) + w(s_j)$  最小原则进行选择;

根据上面所给的目标选择设计程序(参见附录3),运行一次可得到直达,换乘一次和换乘二次的结果为:

##### **(1) S3359 → S1828**

从 S3359 到 S1828 无直达车和地铁

从 S3359 到 S1828,你可以先乘坐 L436 在 S1784 换乘 L167

所需费用为:3 所花时间为:101

L436 行车路线为:S3359->S2026->S1132->S2266->S2263->S3917->S2303->S2301->S3233->S618->S616->S2112->S2110->S2153->S2814->S2813->S3501->S3515->S3500->S756->S492->S903->S1768->S955->S480->S2703->S2800->S2192->S2191->S1829->S3649->S1784

L167 行车路线为:S1784->S1828



从 S3359 到 S1828,你可以先乘坐 L484 在 S2027 换乘 L201 再在 S1790 换乘 L41  
所需费用为:3 所花时间为:73

L484 行车路线为:S3359->S2023->S2027

L201 行车路线为:S2027->S1327->S1842->S609->S483->S604->S2650->S3470->  
S2619->S2340->S2182->S992->S2322->S1770->S1790

L41 行车路线为:S1790->S458->S1792->S1783->S1671->S1828

---

### **(2) S1557 → S481**

从 S1557 到 S481 无直达车和地铁

从 S1557 到 S481,你 cannot 通过换乘一次来达到目的地

从 S1557 到 S481,你可以先乘坐 L457 在 S1920 换乘 L294 再在 S2049 换乘 L516  
所需费用为:3 所花时间为:160

L457 行车路线为:S1557->S3158->S2628->S3408->S2044->S1985->

S2563->S2996->S3567->S884->S2566->S2907->S2313->S3131->S3046->  
S1721->S1920

L294 行车路线为:S1921->S181->S209->S237->S3506->S2534->S2535->S399->  
S1376->S608->S2361->S2057->S614->S626->S889->S2049

L516 行车路线为:S2049->S3709->S607->S3045->S527->S525->S537->S2651->  
S3013->S1808->S1173->S910->S3517->S453->S2424->S1174->S902->S903->  
S2101->S481

---

### **(3) S971 → S485**

从 S971 到 S485 无直达车和地铁

从 S971 到 S485,你可以先乘坐 L13 在 S2184 换乘 L417

所需费用为:3 所花时间为:128

L13 行车路线为:S971->S3832->S3341->S2237->S3565->S3333->S1180->  
S3494->S1523->S1520->S1988->S1743->S1742->S1181->S1879->S3405->  
S2517->S3117->S2954->S531->S2184

L417 行车路线为:S2184->S992->S2322->S1770->S1789->S2119->S2116->  
S3544->S3186->S3409->S2717->S1402->S2840->S643->S2079->S1920->  
S2480->S2482->S2210->S3332->S3351->S485

从 S971 到 S485,你可以先乘坐 L310 在 S1659 换乘 L17 再在 S2082 换乘 L450  
所需费用为:3 所花时间为:127

L310 行车路线为:S971->S3832->S3341->S2237->S3565->S3333->S1180->  
S3494->S1523->S1521->S3755->S1030->S2877->S1774->S1808->S3013->  
S2651->S3693->S1659

L17 行车路线为:S1659->S2962->S622->S456->S427->S582->S577->S1895->  
S3648->S668->S3081->S3078->S2082

L450 行车路线为:S2082->S2213->S2211->S2377->S1495->S1269->S772->  
S72->S771->S485

---

### **(4) S8 → S73**

从 S8 到 S73 无直达车和地铁

从 S8 到 S73,你可以先乘坐 L159 在 S400 换乘 L474

所需费用为:2 所花时间为:80

L159 行车路线为:S8->S3412->S2743->S3586->S2544->S913->S2953->S3874->S630->S854->S400->S2633

L474 行车路线为:S2633->S3053->S410->S411->S2846->S605->S604->S527->S525->S3470->S2619->S2340->S3162->S2181->S2705->S73

从 S8 到 S73,你可以先乘坐 L472 在 S3874 换乘 L231 再在 S1659 换乘 L480

所需费用为:3 所花时间为:82

L472 行车路线为:S8->S3412->S2743->S3586->S2544->S3900->S913->S2953->S3874

L231 行车路线为:S1426->S1427->S630->S854->S88->S609->S483->S604->S2650->S3693->S1659

L480 行车路线为:S1659->S888->S1183->S2619->S2340->S2183->S2181->S73

---

#### **(5) S148 → S485**

从 S148 到 S485 无直达车和地铁

从 S148 到 S485,你不能通过换乘一次来达到目的地

从 S148 到 S485,你可以先乘坐 L308 在 S2388 换乘 L378 再在 S2027 换乘 L469

所需费用为:3 所花时间为:157

L308 行车路线为:S148->S3182->S2215->S474->S2477->S1234->S345->S1419->S3558->S2105->S3835->S1418->S2388

L378 行车路线为:S2388->S1687->S2071->S2068->S1487->S3750->S303->S710->S127->S2312->S3849->S1347->S3885->S1868->S1710->S2363->S3877->S3727->S3697->S1746->S2027

L469 行车路线为:S2027->S2023->S3359->S2026->S1132->S2265->S2654->S1729->S3766->S1691->S1383->S1381->S1321->S2019->S2017->S2159->S772->S485

---

#### **(6) S87 → S3676**

从 S87 到 S3676,你可以乘坐直达车 L231

所需费用为:1 所花时间为:30

行车路线为: S88->S609->S483->S604->S2650->S3693->S1659->S2962->S622->S456->S427

从 S87 到 S3676,你可以先乘坐 L28 在 S608 换乘 L17

所需费用为:2 所花时间为:35

L28 行车路线为:S87->S608

L17 行车路线为:S609->S483->S604->S2650->S3693->S1659->S2962->S622->S456->S427

从 S87 到 S3676,你可以先乘坐 T2 在 D32 换乘 T2 再在 D34 换乘 T2

所需费用为:3 所花时间为:36.5

T2 行车路线为:D27->D28->D29->D30->D31->D32

T2 行车路线为:D32->D18->D33->D34

T2 行车路线为:D34->D35->D36

-----

注：输出结果中对于连不起来的路线表示可以通过地铁站连接各个路线。最后一条实际上给出的换乘两次的最优解实际是条直达线路，这是由于未考虑起始站点和目标站点的换乘时间引起的。可以认为实际上是不建议选择换乘两次的线路的。针对它的算法改进将在第六部分予以考虑。

### 6.2.5 时间优先，费用次优的模型

基于算法设计思路，我们以时间优先，费用次优作为目标来进行求解，每个步骤的目标选择为（符号可参见变量说明）：

Step2 中的目标选择为：根据时间  $t(s_1, s_2)$  最小原则进行选择。如果选择过程中，费用  $t(s_1, s_2)$  相等则按时间  $f(s_1, s_2)$  最小原则进行选择，

Step3 中的目标选择为：根据费用  $t(s_1, s_2) = t(s_1, s_i) + t(s_i, s_2) + u(s_i)$  最小原则进行选择。如果选择过程中，费用  $t(s_1, s_2)$  相等则按时间  $f(s_1, s_2) = f(s_1, s_i) + f(s_i, s_2)$  最小原则进行选择；

Step4 中的目标选择为：根据费用  $t(s_1, s_2) = t(s_1, s_i) + t(s_i, s_j) + t(s_j, s_2) + u(s_i) + u(s_j)$  最小原则进行选择。如果选择过程中，费用  $f(s_1, s_2)$  相等则按时间  $f(s_1, s_2) = f(s_1, s_i) + f(s_i, s_j) + f(s_j, s_2)$  最小原则进行选择；

根据上面所给的目标选择设计程序（参见附录 4），运行一次可得到直达，换乘一次和换乘二次的结果为：

#### **(1) S3359 → S1828**

从 S3359 到 S1828 无直达车和地铁

从 S3359 到 S1828,你可以先乘坐 L436 在 S1784 换乘 L167

所花时间为:101 所需费用为:3

L436 行车路线为:S3359->S2026->S1132->S2266->S2263->S3917->S2303->

S2301->S3233->S618->S616->S2112->S2110->S2153->S2814->S2813->

S3501->S3515->S3500->S756->S492->S903->S1768->S955->S480->

S2703->S2800->S2192->S2191->S1829->S3649->S1784

L167 行车路线为:S1784->S1828

从 S3359 到 S1828,你可以先乘坐 L15 在 S2903 换乘 L201 再在 S1671 换乘 L41

所花时间为:73 所需费用为:3

L15 行车路线为:S3359->S2903

L201 行车路线为:S2903->S2027->S1327->S1842->S609->S483->S604->

S2650->S3470->S2619->S2340->S2182->S992->S2322->S1770->S1790->

S458->S1792->S1783->S1671

L41 行车路线为:S1671->S1828

-----

## **(2) S1557 → S481**

从 S1557 到 S481 无直达车和地铁

从 S1557 到 S481,你 cannot 通过换乘一次来达到目的地

从 S1557 到 S481,你可以先乘坐 L84 在 S1919 换乘 L189 再在 S3186 换乘 L460

所花时间为:106 所需费用为:3

L84 行车路线为:S1557->S3158->S2628->S3408->S2044->S1985->S2563->  
S2682->S28->S29->S55->S51->S1919

L189 行车路线为:S1919->S2840->S1402->S3186

L460 行车路线为:S3186->S3544->S2116->S2119->S1788->S1789->S1770->  
S2322->S992->S2184->S2954->S3117->S2424->S1174->S902->S903->  
S2101->S481

---

## **(3) S971 → S485**

从 S971 到 S485 无直达车和地铁

从 S971 到 S485,你可以先乘坐 L13 在 S2184 换乘 L417

所花时间为:128 所需费用为:3

L13 行车路线为:S971->S3832->S3341->S2237->S3565->S3333->S1180->  
S3494->S1523->S1520->S1988->S1743->S1742->S1181->S1879->S3405->  
S2517->S3117->S2954->S531->S2184

L417 行车路线为:S2184->S992->S2322->S1770->S1789->S2119->S2116->  
S3544->S3186->S3409->S2717->S1402->S2840->S643->S2079->S1920->  
S2480->S2482->S2210->S3332->S3351->S485

从 S971 到 S485,你可以先乘坐 L94 在 S567 换乘 T1 再在 S466 换乘 L51

所花时间为:93 所需费用为:5

L94 行车路线为:S971->S3571->S1609->S345->S1419->S2389->S567

D1 行车路线为:D1->D2->D3->D4->D5->D6->D7->D8->D9->D10->D11->  
D12->D13->D14->D15->D16->D17->D18->D19->D20->D21

L51 行车路线为:S466->S3189->S2810->S2385->S71->S485

---

## **(4) S8 → S73**

从 S8 到 S73 无直达车和地铁

从 S8 到 S73,你可以先乘坐 L159 在 S400 换乘 L474

所花时间为:80 所需费用为:2

L159 行车路线为:S8->S3412->S2743->S3586->S2544->S913->S2953->S3874->  
S630->S854->S400->S2633

L474 行车路线为:S2633->S3053->S410->S411->S2846->S605->S604->S527->  
S525->S3470->S2619->S2340->S3162->S2181->S2705->S73

从 S8 到 S73,你可以先乘坐 L200 在 S2534 换乘 T1 再在 S609 换乘 L57

所花时间为:62.5 所需费用为:5

L200 行车路线为:S8->S3412->S2743->S2544->S2953->S778->S2534

D1 行车路线为:D15->D14->D13->D12

L57 行车路线为:S609->S483->S604->S2650->S3470->S2619->S2340->  
S3162->S2181->S73

---

### **(5) S148 → S485**

从 S148 到 S485 无直达车和地铁

从 S148 到 S485,你 cannot 通过换乘一次来达到目的地

从 S148 到 S485,你可以先乘坐 L24 在 S1487 换乘 T1 再在 S466 换乘 L50  
所花时间为:87.5 所需费用为:5

L24 行车路线为:S148->S927->S2830->S2070->S1487

D1 行车路线为:D2->D3->D4->D5->D6->D7->D8->D9->D10->D11->D12->  
D13->D14->D15->D16->D17->D18->D19->D20->D21

L50 行车路线为:S466->S964->S3189->S2810->S2385->S71->S485

---

### **(6) S87 → S3676**

从 S87 到 S3676,你可以乘坐地铁 T2

所花时间为:25 所需费用为:3

行车路线为: D27->D28->D29->D30->D31->D32->D18->D33->D34->D35->D36

从 S87 到 S3676,你可以先乘坐 T2 在 D28 换乘 T2

所花时间为:29 所需费用为:3

D2 行车路线为:D27->D28

D2 行车路线为:D28->D29->D30->D31->D32->D18->D33->D34->D35->D36

从 S87 到 S3676,你可以先乘坐 T2 在 D28 换乘 T2 再在 S582 换乘 L381

所花时间为:35.5 所需费用为:4

D2 行车路线为:D27->D28

D2 行车路线为:D28->D29->D30->D31->D32->D18->D33->D34->D35

L381 行车路线为:S582->S427

---

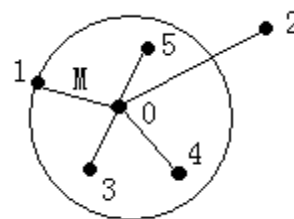
注: 输出结果中对于连不起来的路线表示可以通过地铁站连接各个路线。对于最后一个算例给出的换乘两次的最优解实际是一条换乘一次的线路,同样的,可以认为是系统不建议乘客换乘两次到达目标站点。

## **6.3 问题三**

### **6.3.1 增广站点概念引入**

对于问题三,多了步行的线路选择方式。那么,步行的话不用沿着公汽或地铁的路线走,任何两个站点之间都可以步行到达。但是,每个人步行也总有个限度,每次步行都应该有个最大步行时限,设为  $M$ 。本文认为超过这个限制,则乘客不再选择走路。由于步行方式的选择,换乘的范围进一步被扩大,本来不是在同一汽车站停靠的,也没有因地铁站而联系起来的两个汽车站点,都可以通过步行,走过去换乘。

这里,我们比前面提到的广义站点更进一步地提出增广站点的概念。在这一问中,对于一个站点,若要在 这个



站点换乘，则对应的站点是个增广站点。把包括且仅包括步行可达的站点的范围用增广站点来表示。在这个步行范围内，所有的站点等价，即可以换乘任何一辆在这些站点停靠的交通线路（包括公汽和地铁）。例如右图：

对于站点 0 而言，它和 1,2,3,4,5 站点原本都是不连通的。但考虑了步行之后，只要走的时间少于或等于  $M$ ，则可以把它们看作是可以换乘的。就如右图中的 1,3,4,5 站点只要走少于或等于  $M$  的时间就能到，能换乘，而走到站点 2 的时间要大于  $M$ ，则画圈的范围可以作为一个增广站点。这个站点的换乘线路包括了经过站点 0,1,3,4,5 的所有线路。这里所指的线路包括了公汽线路和地铁线路。而站点 0,1,3,4,5 本身也完全可能是广义站点。若是，则要在计算走路时间时也得算上在广义站点里的行走时间。

这样，与之平行的，可以认为这个增广站点里的每个站点就都等价了，在这个增广站点可以换乘的线路是经过它包含的实际站点的线路并集。

当然，增广站点的换乘时间  $w$  取值就不是 5，也不是 4 种取值，而有五种取法，无数种值。因为有五种不同的换乘方式，所以  $w$  的取值如下：

$$w = \begin{cases} 5, & \text{进站和出站都是公汽} \\ 4, & \text{进站和出站都是地铁} \\ 7, & \text{进站是地铁，出站是公汽} \\ 6, & \text{进站是公汽，出站是地铁} \\ t_{\text{等}} + t_{\text{走}}, & \text{通过步行到其它站点换乘} \end{cases}$$

其中  $t_{\text{等}}$  表示换乘交通工具所要等待的时间，等于在同站点换乘的时间减去同站点步行时间

$$t_{\text{等}} = \begin{cases} 3, & \text{等待公汽时间} \\ 2, & \text{等待地铁时间} \end{cases}$$

而  $t_{\text{走}}$  表示从增广站点中的一个实际站点步行到另一个站点的时间。这个时间是题目中的数据没有给出，也是因人而异的。有的人可能走得快一点，或是知道近路，而有的人走得慢，或绕远路等等，导致时间有所不同。

需要特别提出的是，这个增广站点的大小是可以变化的。这个也很好理解，步行的极限时间越大，乘客越愿意走更多的路去乘交通工具，那么增广站点的范围也就越大，所包含的站点也就越多，则在这个增广站点能换乘的线路也就越多，这也是称其为增广站点的原因。而相反的  $M$  越小，增广站点的范围也就越小，所包含的站点也就越少。在极端的情况下，若  $M=0$ ，则表示乘客不愿走一步路，那么问题自动转化为第二问，而本模型也自动退化为第二问的模型。

### 6.3.2 数学模型

根据上述处理，也可以得到问题三的数学模型如下：

对于任意两个站点  $s_0$  和  $s_n$ （这里的站点和以下的站点都可以是增广站点），设以  $s_0$  为起始点， $s_n$  为目的地的路线有  $m$  条，路线集合  $C = \{c_1, c_2, \dots, c_m\}$ ，其中任意一条路线  $c_i = (s_0, s_1, s_2, \dots, s_{n-1}, s_n)$ ，路线  $c_i$  中有  $q$  个站点换乘，分别为  $s_{j1}, s_{j2}, \dots, s_{jq}$ ，由于假设已经知道所有站点之间的步行时间，此时还需要同时考虑公汽线路与地铁线路，则  $f(s_i, s_j)$  与  $g(s_i, s_j)$  的关系如下：

$$f(s_i, s_j) = \begin{cases} 3, & \text{若乘地铁} \\ 1, & \text{若乘汽车单一计价} \\ 1, & \text{若乘汽车, 且 } 1 \leq g(s_i, s_j) \leq 20 \\ 2, & \text{若乘汽车, 且 } 21 \leq g(s_i, s_j) \leq 40 \\ 3, & \text{若乘汽车, 且 } g(s_i, s_j) \geq 41 \end{cases}$$

$$\text{且 } t(s_k, s_{k+1}) = \begin{cases} 3, & \text{如果在相邻站点 } s_k \text{ 与 } s_{k+1} \text{ 通过乘汽车到达} \\ 2.5, & \text{如果在相邻站点 } s_k \text{ 与 } s_{k+1} \text{ 通过乘地铁到达} \end{cases}$$

$$u(s_i) = \begin{cases} 5, & \text{在该站点处公汽换乘公汽} \\ 4, & \text{在该站点处地铁换乘地铁} \\ 7, & \text{在该站点处地铁换乘公汽} \\ 6, & \text{在该站点处公汽换乘地铁} \\ t_{\text{等}}(s_i) + t_{\text{走}}(s_i), & \text{在增广结点 } s_i \text{ 处通过步行到其它站点换乘} \end{cases}$$

其中  $t_{\text{等}}(s_i)$  表示在增广结点  $s_i$  处换乘交通工具所要等待的时间,

$$t_{\text{等}}(s_i) = \begin{cases} 3, & \text{等待公汽时间} \\ 2, & \text{等待地铁时间} \end{cases},$$

$t_{\text{走}}(s_i)$  表示从增广结点  $s_i$  的一个站点步行到另一个站点的时间, 该值由题目中假设已知。

则路线  $c_i$  上所需费用

$$F(c_i) = \begin{cases} f(s_0, s_n), & q = 0 \\ f(s_0, s_{j1}) + f(s_{j1}, s_n), & q = 1 \\ f(s_0, s_{j1}) + \sum_{k=1}^{q-1} f(s_{jk}, s_{j(k+1)}) + f(s_{jq}, s_n), & q \geq 2 \end{cases}$$

$$\text{令决策变量 } x_i = \begin{cases} 1, & \text{在站点 } s_i \text{ 换乘} \\ 0, & \text{否则} \end{cases}, \text{ 则 } q = \sum_{i=1}^{n-1} x_i$$

$$\text{路线上 } c_i \text{ 所花费的时间 } T(c_i) = \sum_{k=0}^{n-1} t(s_k, s_{k+1}) + \sum_{i=1}^{n-1} x_i u(s_i)$$

设  $A = \{L_1, L_2, \dots, L_h\}$  表示经过起始点  $s_0$  的所有交通工具的集合,  $B = \{L'_1, L'_2, \dots, L'_u\}$  表示到达目的地  $s_n$  的所有交通工具的集合,

$$\text{令 } y_i = \begin{cases} 1, & \text{在起始点 } s_0 \text{ 处选择交通工具 } L_i \\ 0, & \text{否则} \end{cases}, \quad z_i = \begin{cases} 1, & \text{选择交通工具 } L'_i \text{ 到达目的地 } s_n \\ 0, & \text{否则} \end{cases}$$

则分别以花费时间最小和所需费用最小为目标, 建立多目标规划模型:

$$\begin{aligned} \min & \quad T(c_i) \\ \min & \quad F(c_i) \quad , \quad \forall c_i \in C \end{aligned}$$

$$s.t. \quad \begin{cases} \sum_{i=1}^{n-1} x_i \leq 2 \\ \sum_{i=1}^h y_i = 1 \\ \sum_{i=1}^u z_i = 1 \end{cases}$$

### 6.3.3 算法描述

把增广站点应用到问题中，给出本题的具体算法如下：

STEP0: 根据前文中提出的数据构架建立公汽站点数据库。同时，以矩阵的形式保存站点之间的步行时间，其中元素的值表示任意两实际站点间行走时间。

STEP1: 输入起始站点  $A$  和目标站点  $B$  以及步行极限时间  $M$ ；

STEP2: 对于起始站点  $A$  与目标站点  $B$  判断有无直达线路（由于此判断程序需要多次调用，具体的判断程序的算法在整体算法后附上，与前文中 *Function* 函数调用相同，但算法不同）：若有唯一的一条直达车则输出，若有不止一条，则按评价条件和评价的关键字优先等级取最优的输出；

STEP3: 判断起始站点  $A$  和目标站点  $B$  有无一次换乘路线：

Step3.1 从公汽站点数据库中查出经过起始站点  $A$  的线路  $L_i (i=1,2,\dots,m)$

Step3.2 查出公汽线路  $L_i$  经过的站点  $G_{ig}$ ，对于每一个站点  $G_{ig}$  与目标站点  $B$  判断有无直达线路：若有，记乘车路线为  $R$ 。计算时间和费用，与当前经一次换乘到达的最佳路线  $R^*$  比较，若当前还没有经一次换乘到达的最佳路线，即  $R^* = \emptyset$ ，或本次查到的路线比当前最佳路线更优，则将本次查到的路线作为最佳路线，即  $R \rightarrow R^*$ ；否则，不作处理；

Step3.3 若存在一次换乘到达的最佳路线  $R^*$ ，即  $R^* \neq \emptyset$ ，则输出  $R^*$ ；

STEP4: 判断起始站点  $A$  和目标站点  $B$  有无两次换乘路线：

Step4.1 从公汽站点数据库中查出经过起始站点  $A$  的线路  $L_i (i=1,2,\dots,m)$  和 经过目标站点  $B$  的公汽线路  $S_j (j=1,2,\dots,n)$ ；

Step4.2 查出公汽线路  $L_i$  经过的站点  $G_{ig}$  和  $S_j$  经过的站点  $H_{jh}$ ，对于所有站点  $G_{ig}$  与所有站点  $H_{jh}$  判断有无直达线路：若有，记乘车路线为  $R$ 。计算时间和费用，与当前经一次换乘到达的最佳路线  $R^*$  比较，若  $R^* = \emptyset$ ，或  $R$  比  $R^*$  更优，则将本次查到的路线作为最佳路线，即  $R \rightarrow R^*$ ；否则，不作处理；

Step4.3 若存在两次换乘到达的最佳路线  $R^*$ ，即  $R^* \neq \emptyset$ ，则输出  $R^*$ ；

STEP5: 若前面的步骤有输出，则停止，否则，报告无合适路线。

对于本文提到的判断两站点  $X$  与  $Y$  之间是否有直达车的子程序的算法描述如下：

Step1: 查出与站点  $X$  相距小于或等于  $M$  的站点  $U_i (i=1,2,\dots,n)$ ，其中包括站点  $X$  本身，查出与站点  $Y$  相距小于或等于  $M$  的站点  $V_j (j=1,2,\dots,n)$ ，其中包括站点  $Y$  本身；

step2: 查出经过站点  $U_i$  的所有线路  $G_{ig}$ ，查出经过站点  $V_j$  的所有线路  $H_{jh}$ ；

step3: 判断有无  $G_{ig} = H_{jh}$ ，若有，则返回真，若没有，则返回假。

由于增广站点的引入，这个算法得以实现。它其实与前两问中的算法有相同的框架。最主要的区别就在于判断两站点是否直达时应用了增广站点，等价了一定范围内的实际站点。这样不仅使前两问的算法得以继续沿用，也使得程序实现的变动幅度得到了有效



降低。同时，寻到的解也能保证是在规定了换乘次数，限定了步行极限，按一定目标求得的最优解。

## 七、模型评价与改进方向

本文建立的每一问的模型之间是相互递进的，它们都是向下兼容的。第二问的模型兼容了第一问，第三问的模型兼容了第二问。这之间的递进通过新的概念和处理方式的引入得以实现。并且所有的模型都给出了它的算法，算法之间也是相同的递进关系。

这里建立的算法都是搜索算法，成功地实现了一个子程序的多次调用，降低了编程的难度，同时也使效率得到了提高，对于本题要求的 6 个算例程序运行的平均时间在 30 秒左右。其中，仅在第 4 个和第 6 个算例中运行时间较长，其它都运行的很快。这与数据的好坏有关系，而不是算法本身存在的问题。不过，在算法的编程具体实现时，由于交通工具的类型不同，包括行车方式，收费方式等都有很大的不同，还是需要在判断时进行严密考虑。不过算法给出了在三种换乘条件下，不同目标的最优路线选择，为乘客提供了更多的选择。

不过，本文设计的算法在涉及地铁站时，假设了不考虑在起始站点和目标站点的换乘时间。而其实，这部分时间还是应该要计算在内的。可以通过增加判断起始站点和目标站点的特性，若有步行去其它站点的现象，则对它进行特殊处理，预先计一次换乘时间，但不累计换乘次数，或者增加人工干预来解决问题。

另外，本文假设了换乘次数不大于两次，但如果起始站点和目标站点相距甚远或由于其他原因确实不能通过换乘两次到达，而乘客又确实要乘交通工具去那里，这时如果还用搜索算法，可能考虑的站点会相当多，导致运行时间难以忍受，这时可以尝试采取蚁群算法、遗传算法等智能算法。

## 八、模型扩展

(1) 本文在模型中提出两个目标，一个是时间最短，一个是费用最小，然后按目标的主次之分，即先满足主要目标，然后再满足次要目标，来建立两个模型。把目标按主次之分时，其实已经默认这两个目标是不同的等级，它们的优先级不同。若考虑两个目标是同一优先级的，则此时可以对这两个目标赋以权值，然后以加权和来作为目标函数，此时本文算法依然有效，只需在比较路线时依据该目标来进行比较即可。

(2) 在本文的模型中，假设同类交通工具相邻站点的间距是相同的。其实，当它们不同时，本文的模型还是适用的。但是在考虑站点间的行进时间将不同，不再是 3 和 2.5 这两种值，而是视具体的行车时间而赋予的不同值。在应用算法求解的实现中需要把每一段路所花的时间一个一个累加起来，而不能通过统计过了多少站，再乘上时间了事。这会导致编程实现上增加了一定的难度，但是这并不影响算法的正确性。

(3) 本文所设计的模型和算法都是针对确定量设计的，无法直接适用于随机量的模型。但是，考虑到现实生活中更普遍的情况，汽车和地铁都不是匀速的，人步行的速度也都不是匀速的，更有上下班高峰等情况，导致行车时间和换乘时间其实都是一个随机量。这时，本文的模型中所设的值就都变成了随机量，本文中的算法不能直接适用。但采用随机模拟的方法，把这些量都模拟出来后，这些值可以再次变回确定量，本文的模型和算法就可以再度适用了。

## 参考文献

- [1]Stanley B.Lippman 等著，李师贤等译，C++ Primer 中文版，北京：人民邮电出版社，2006 年 3 月  
[2]Charles K.Alexander 等著，刘巽亮等译，电路基础，北京：电子工业出版社，2003 年  
[3]严蔚敏 吴伟民，数据结构(C 语言版)，北京：清华大学出版社，1997 年 4 月  
[4]张存宝 李华 严新平，基于 Web GIS 的城市公交问路系统，武汉理工大学学报，2004.2 Vol.28 No.1:99~102

## 附录

### 附录1

```
#include <iostream>
#include <fstream>
#include <vector>
#include <iomanip>
#include <string>
using namespace std;

ifstream infile("data.txt", ifstream::in);
ofstream outfile("result.txt", ofstream::out);

typedef struct
{
    char F_B;
    int i;
    int num;
}car;

vector< vector<car> > station(3958);

struct node
{
    vector<int> forward;
    vector<int> backward;
    char R_Y;
    char X_FB_C;
}Car[521];

const int Max=1000;
const int CarRunTime=3;
const int CarCarTime=5;

int up, down;
int car1, car2, car3;
int charge1, charge2, charge3;
char fb1, fb2, fb3;
int time1, time2, time3;
int at, at1, at2;
```

```

void Data_Input()
{
    char l, ch;
    int car_i, zd, i, j;
    car c;
    infile>>l;
    while(infile>>car_i) {
        infile>>Car[car_i].R_Y;
        infile>>Car[car_i].X_FB_C;
        if(Car[car_i].X_FB_C=='X') {
            i=0;
            while(infile>>ch&&ch=='S') {
                infile>>zd;
                Car[car_i].forward.push_back(zd);
                c.i=i++;
                c.num=car_i;
                c.F_B='F';
                station[zd].push_back(c);
            }
            for(i--, j=0; i>=0; i--, j++) {
                zd=Car[car_i].forward[i];
                Car[car_i].backward.push_back(zd);
                c.num=car_i;
                c.i=j;
                c.F_B='B';
                station[zd].push_back(c);
            }
        }
        else if(Car[car_i].X_FB_C=='F') {
            i=0;
            while(infile>>ch&&ch=='S') {
                infile>>zd;
                Car[car_i].forward.push_back(zd);
                c.i=i++;
                c.num=car_i;
                c.F_B='F';
                station[zd].push_back(c);
            }
            i=0;
            while(infile>>ch&&ch=='S') {
                infile>>zd;
                Car[car_i].backward.push_back(zd);
                c.i=i++;
                c.num=car_i;
                c.F_B='B';
                station[zd].push_back(c);
            }
        }
        else if(Car[car_i].X_FB_C=='C') {
            i=0;

```

```

        while(infile>>ch&&ch=='S'){
            infile>>zd;
            Car[car_i].forward.push_back(zd);
            c.i=i++;
            c.num=car_i;
            c.F_B='F';
            station[zd].push_back(c);
        }
    }
    if(ch=='E'){
        break;
    }
}

bool Go_Input()
{
    string u,d;
    cout<<"请输入起始站点和目的站点:";
    cin>>u>>d;
    up=1000*(u[1]-'0')+100*(u[2]-'0')+10*(u[3]-'0')+(u[4]-'0');
    down=1000*(d[1]-'0')+100*(d[2]-'0')+10*(d[3]-'0')+(d[4]-'0');
    return false;
}

void Output0(int up,int down,int chooose,char fb)
{
    int i,j,k;
    for(i=0; i<station[up].size(); i++)
        if(station[up][i].num==chooose&&station[up][i].F_B==fb)
            break;
    for(j=0; j<station[down].size(); j++)
        if(station[down][j].num==chooose&&station[down][j].F_B==fb)
            break;
    if(fb=='F'){
        for(k=station[up][i].i; k<=station[down][j].i-1; k++)
            outfile<<"S"<<Car[station[up][i].num].forward[k]<<"->";
        outfile<<"S"<<Car[station[up][i].num].forward[k];
    }
    else if(fb=='B'){
        for(k=station[up][i].i; k<=station[down][j].i-1; k++)
            outfile<<"S"<<Car[station[up][i].num].backward[k]<<"->";
        outfile<<"S"<<Car[station[up][i].num].backward[k];
    }
}

int RY(char ry,int t)
{
    if(ry=='R'){
        if(t<=20)
            return 1;
    }
}

```

```

        else if(t<=40)
            return 2;
        else
            return 3;
    }
    return 1;
}

void Get1(int c1,int c2,char f1,char f2,int a){
    car1=c1;car2=c2;
    fb1=f1;fb2=f2;
    at=a;
}

void Get2(int c1,int c2,int c3,char f1,char f2,char f3,int a1,int a2)
{
    car1=c1;car2=c2;car3=c3;
    fb1=f1;fb2=f2;fb3=f3;
    at1=a1;at2=a2;
}

bool Change0(int up,int down,int &choose,int &charge,int &time,char &fb)
{
    int i,j,t;
    for(i=0; i<station[up].size(); i++){
        for(j=0; j<station[down].size(); j++){
            if(station[up][i].num==station[down][j].num&&station[up][i].F_B==station[down][j].
F_B&&station[up][i].i<station[down][j].i){
                t=station[down][j].i-station[up][i].i;
                if(Car[station[up][i].num].R_Y=='R'){
                    if(t<=20){
                        if(charge>1){
                            charge=1;
                            time=t*CarRunTime;
                            choose=station[up][i].num;
                            fb=station[up][i].F_B;
                        }
                        else if(charge==1&&t*CarRunTime<time){
                            time=t*CarRunTime;
                            choose=station[up][i].num;
                            fb=station[up][i].F_B;
                        }
                    }
                }
            }
            else if(t<=40){
                if(charge>2){
                    charge=2;
                    time=t*CarRunTime;
                    choose=station[up][i].num;
                    fb=station[up][i].F_B;
                }
                else if(charge==2&&t*CarRunTime<time){

```

```

        time=t*CarRunTime;
        choose=station[up][i].num;
        fb=station[up][i].F_B;
    }
}
else{
    if(charge>3){
        charge=3;
        time=t*CarRunTime;
        choose=station[up][i].num;
        fb=station[up][i].F_B;
    }
    else if(charge==3&&time<t*CarRunTime){
        time=t*CarRunTime;
        choose=station[up][i].num;
        fb=station[up][i].F_B;
    }
}
}
else{
    if(charge>1){
        charge=1;
        time=t*CarRunTime;
        choose=station[up][i].num;
        fb=station[up][i].F_B;
    }
    else if(charge==1&&time<t*CarRunTime){
        time=t*CarRunTime;
        choose=station[up][i].num;
        fb=station[up][i].F_B;
    }
}
}
}
}
if(charge==Max)
    return false;
return true;
}

void Changel()
{
    int i, j, choose, charge=Max, time=Max;
    char fb;
    for(i=0; i<station[up].size(); i++){
        if(station[up][i].F_B=='F'){
            for(j=station[up][i].i+1; j<Car[station[up][i].num].forward.size(); j++){
                charge2=time2=Max;

                if(Change0(Car[station[up][i].num].forward[j], down, choose, charge2, time2, fb)){
                    time1=(j-station[up][i].i)*CarRunTime;

```

```

        charge1=R_Y(Car[station[up][i].num].R_Y, j-station[up][i].i);
        if(charge1+charge2<charge) {
            charge=charge1+charge2;
            time=time1+time2;

        Get1(station[up][i].num, choose, 'F', fb, Car[station[up][i].num].forward[j]);
        }
        else if(charge1+charge2==charge&&time1+time2<time) {
            time=time1+time2;

        Get1(station[up][i].num, choose, 'F', fb, Car[station[up][i].num].forward[j]);
        }
    }
}
else if(station[up][i].F_B=='B') {
    for(j=station[up][i].i+1; j<Car[station[up][i].num].backward.size(); j++) {
        charge2=time2=Max;

        if(Change0(Car[station[up][i].num].backward[j], down, choose, charge2, time2, fb)) {
            time1=(j-station[up][i].i)*CarRunTime;
            charge1=R_Y(Car[station[up][i].num].R_Y, j-station[up][i].i);
            if(charge1+charge2<charge) {
                charge=charge1+charge2;
                time=time1+time2;

            Get1(station[up][i].num, choose, 'B', fb, Car[station[up][i].num].backward[j]);
            }
            else if(charge1+charge2==charge&&time1+time2<time) {
                time=time1+time2;

            Get1(station[up][i].num, choose, 'B', fb, Car[station[up][i].num].backward[j]);
            }
        }
    }
}
if(charge!=Max) {
    outfile<<"从S"<<up<<"到S"<<down<<"，你可以先乘坐L"<<car1<<"在S"<<at<<"换乘
L"<<car2<<endl;
    outfile<<"所需费用为："<<charge<<ends<<"所花时间为："<<time+CarCarTime<<endl;
    outfile<<"L"<<car1<<"行车路线为：";
    Output0(up, at, car1, fb1);
    outfile<<endl<<"L"<<car2<<"行车路线为：";
    Output0(at, down, car2, fb2);
    outfile<<endl<<endl;
} else
    outfile<<"从S"<<up<<"到S"<<down<<"，你 cannot 通过换乘一次车来达到目的地
"<<endl<<endl;
}

```

```

void Change2()
{
    int i, j, k, h, choose, charge=Max, time=Max;
    char fb;
    for(i=0; i<station[up].size(); i++) {
        if(station[up][i].F_B=='F') {
            for(j=station[up][i].i+1; j<Car[station[up][i].num].forward.size(); j++) {
                for(k=0; k<station[down].size(); k++) {
                    if(station[down][k].F_B=='F') {
                        for(h=station[down][k].i-1; h>=0; h--) {
                            charge2=time2=Max;

                            if(Change0(Car[station[up][i].num].forward[j], Car[station[down][k].num].forward[h],
                                choose, charge2, time2, fb)) {

                                time1=(j-station[up][i].i)*CarRunTime;
                                charge1=RY(Car[station[up][i].num].R_Y, j-
station[up][i].i);

                                time3=(station[down][k].i-h)*CarRunTime;

                                charge3=RY(Car[station[down][k].num].R_Y, station[down][k].i-h);
                                if(charge1+charge2+charge3<charge) {
                                    charge=charge1+charge2+charge3;
                                    time=time1+time2+time3;

                                    Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'F', Car[station[up][i].
num].forward[j], Car[station[down][k].num].forward[h]);
                                }
                                else
                                if(charge1+charge2+charge3==charge&&time1+time2+time3<time) {
                                    time=time1+time2+time3;

                                    Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'F', Car[station[up][i].
num].forward[j], Car[station[down][k].num].forward[h]);
                                }
                            }
                        }
                    }
                }
            }
        }
        else if(station[down][k].F_B=='B') {
            for(h=station[down][k].i-1; h>=0; h--) {
                charge2=time2=Max;

                if(Change0(Car[station[up][i].num].forward[j], Car[station[down][k].num].backward[h],
                    choose, charge2, time2, fb)) {

                    time1=(j-station[up][i].i)*CarRunTime;
                    charge1=RY(Car[station[up][i].num].R_Y, j-
station[up][i].i);

                    time3=(station[down][k].i-h)*CarRunTime;

                    charge3=RY(Car[station[down][k].num].R_Y, station[down][k].i-h);
                    if(charge1+charge2+charge3<charge) {
                        charge=charge1+charge2+charge3;

```





```

        for(h=station[down][k].i-1; h>=0; h--) {
            charge2=time2=Max;

            if(Change0(Car[station[up][i].num].backward[j], Car[station[down][k].num].backward[
h], choose, charge2, time2, fb)) {
                time1=(j-station[up][i].i)*CarRunTime;
                charge1=RY(Car[station[up][i].num].R_Y, j-
station[up][i].i);
                time3=(station[down][k].i-h)*CarRunTime;

                charge3=RY(Car[station[down][k].num].R_Y, station[down][k].i-h);
                if(charge1+charge2+charge3<charge) {
                    charge=charge1+charge2+charge3;
                    time=time1+time2+time3;

                    Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'B', Car[station[up][i].
num].backward[j], Car[station[down][k].num].backward[h]);
                }
                else
if(charge1+charge2+charge3==charge&&time1+time2+time3<time) {
                    time=time1+time2+time3;

                    Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'B', Car[station[up][i].
num].backward[j], Car[station[down][k].num].backward[h]);
                }
            }
        }
    }
}

if(charge!=Max) {
    outfile<<"从S"<<up<<"到S"<<down<<"，你可以先乘坐L"<<car1<<"在S"<<at1<<"换乘
L"<<car2<<"后再在S"<<at2<<"换乘L"<<car3<<endl;
    outfile<<"所需费用为："<<charge<<ends<<"所花时间为："<<time+CarCarTime*2<<endl;
    outfile<<"L"<<car1<<"行车路线为：";
    Output0(up, at1, car1, fb1);
    outfile<<endl<<"L"<<car2<<"行车路线为：";
    Output0(at1, at2, car2, fb2);
    outfile<<endl<<"L"<<car3<<"行车路线为：";
    Output0(at2, down, car3, fb3);
    outfile<<endl<<endl;
} else
    outfile<<"从S"<<up<<"到S"<<down<<"，你不能通过换乘二次车来达到目的地
"<<endl<<endl;
}

int main()
{

```

```

Data_Input();
Go_Input();
int choose, charge=Max;
char fb;
int time=Max;
if(Change0(up, down, choose, charge, time, fb)) {
    outfile<<"从S"<<up<<"到S"<<down<<"，你可以乘坐直达车L"<<choose<<endl;
    outfile<<"所需费用为:"<<charge<<ends<<"所花时间为:"<<time<<endl;
    outfile<<"行车路线为: ";
    Output0(up, down, choose, fb);
    outfile<<endl;
}
else{
    outfile<<"从S"<<up<<"到S"<<down<<"无直达车"<<endl<<endl;
}
Change1();
Change2();
return 0;
}

```

## 附录2

```

#include <iostream>
#include <fstream>
#include <vector>
#include <iomanip>
#include <string>
using namespace std;

ifstream infile("data.txt", ifstream::in);
ofstream outfile("result.txt", ofstream::out);

typedef struct
{
    char F_B;
    int i;
    int num;
}car;

vector< vector<car> > station(3958);

struct node
{
    vector<int> forward;
    vector<int> backward;
    char R_Y;
    char X_FB_C;
}Car[521];

const int Max=1000;
const int CarRunTime=3;
const int CarCarTime=5;

```

```

int up, down;
int car1, car2, car3;
int charge1, charge2, charge3;
char fb1, fb2, fb3;
int time1, time2, time3;
int at, at1, at2;

void Data_Input()
{
    char l, ch;
    int car_i, zd, i, j;
    car c;
    infile>>l;
    while(infile>>car_i) {
        infile>>Car[car_i].R_Y;
        infile>>Car[car_i].X_FB_C;
        if(Car[car_i].X_FB_C=='X') {
            i=0;
            while(infile>>ch&&ch=='S') {
                infile>>zd;
                Car[car_i].forward.push_back(zd);
                c.i=i++;
                c.num=car_i;
                c.F_B='F';
                station[zd].push_back(c);
            }
            for(i--, j=0; i>=0; i--, j++) {
                zd=Car[car_i].forward[i];
                Car[car_i].backward.push_back(zd);
                c.num=car_i;
                c.i=j;
                c.F_B='B';
                station[zd].push_back(c);
            }
        }
        else if(Car[car_i].X_FB_C=='F') {
            i=0;
            while(infile>>ch&&ch=='S') {
                infile>>zd;
                Car[car_i].forward.push_back(zd);
                c.i=i++;
                c.num=car_i;
                c.F_B='F';
                station[zd].push_back(c);
            }
            i=0;
            while(infile>>ch&&ch=='S') {
                infile>>zd;
                Car[car_i].backward.push_back(zd);
                c.i=i++;

```

```

        c.num=car_i;
        c.F_B='B';
        station[zd].push_back(c);
    }
}
else if(Car[car_i].X_FB_C=='C'){
    i=0;
    while(infile>>ch&&ch=='S'){
        infile>>zd;
        Car[car_i].forward.push_back(zd);
        c.i=i++;
        c.num=car_i;
        c.F_B='F';
        station[zd].push_back(c);
    }
}
if(ch=='E'){
    break;
}
}
}

bool Go_Input()
{
    string u,d;
    cout<<"请输入起始站点和目的站点:";
    cin>>u>>d;
    up=1000*(u[1]-'0')+100*(u[2]-'0')+10*(u[3]-'0')+(u[4]-'0');
    down=1000*(d[1]-'0')+100*(d[2]-'0')+10*(d[3]-'0')+(d[4]-'0');
    return false;
}

void Output0(int up,int down,int chooose,char fb)
{
    int i,j,k;
    for(i=0; i<station[up].size(); i++)
        if(station[up][i].num==chooose&&station[up][i].F_B==fb)
            break;
    for(j=0; j<station[down].size(); j++)
        if(station[down][j].num==chooose&&station[down][j].F_B==fb)
            break;
    if(fb=='F'){
        for(k=station[up][i].i; k<=station[down][j].i-1; k++)
            outfile<<"S"<<Car[station[up][i].num].forward[k]<<"->";
        outfile<<"S"<<Car[station[up][i].num].forward[k];
    }
    else if(fb=='B'){
        for(k=station[up][i].i; k<=station[down][j].i-1; k++)
            outfile<<"S"<<Car[station[up][i].num].backward[k]<<"->";
        outfile<<"S"<<Car[station[up][i].num].backward[k];
    }
}

```

```

}

int RY(char ry, int t)
{
    if(ry=='R') {
        if(t<=20)
            return 1;
        else if(t<=40)
            return 2;
        else
            return 3;
    }
    return 1;
}

void Get1(int c1, int c2, char f1, char f2, int a) {
    car1=c1;car2=c2;
    fb1=f1;fb2=f2;
    at=a;
}

void Get2(int c1, int c2, int c3, char f1, char f2, char f3, int a1, int a2)
{
    car1=c1;car2=c2;car3=c3;
    fb1=f1;fb2=f2;fb3=f3;
    at1=a1;at2=a2;
}

bool Change0(int up, int down, int &chooose, int &charge, int &time, char &fb)
{
    int i, j, t;
    for(i=0; i<station[up].size(); i++) {
        for(j=0; j<station[down].size(); j++) {

            if(station[up][i].num==station[down][j].num&&station[up][i].F_B==station[down][j].
F_B&&station[up][i].i<station[down][j].i) {
                t=station[down][j].i-station[up][i].i;
                if(t*CarRunTime<time) {
                    time=t*CarRunTime;
                    charge=RY(Car[station[up][i].num].R_Y, t);
                    chooose=station[up][i].num;
                    fb=station[up][i].F_B;
                }
                else if(t*CarRunTime==time) {
                    if(Car[station[up][i].num].R_Y=='R') {
                        if(t<=20&&charge>1) {
                            charge=1;
                            chooose=station[up][i].num;
                            fb=station[up][i].F_B;
                        }
                        else if(t<=40&&charge>2) {

```

```

        charge=2;
        chooose=station[up][i].num;
        fb=station[up][i].F_B;
    }
    else if(charge>3) {
        charge=3;
        chooose=station[up][i].num;
        fb=station[up][i].F_B;
    }
}
else{
    if(charge>1) {
        charge=1;
        chooose=station[up][i].num;
        fb=station[up][i].F_B;
    }
}
}
}
}
}
}
if(time==Max)
    return false;
return true;
}

void Change1()
{
    int i, j, choose, charge=Max, time=Max;
    char fb;
    for(i=0; i<station[up].size(); i++) {
        if(station[up][i].F_B=='F') {
            for(j=station[up][i].i+1; j<Car[station[up][i].num].forward.size(); j++) {
                charge2=time2=Max;

                if(Change0(Car[station[up][i].num].forward[j], down, choose, charge2, time2, fb)) {
                    time1=(j-station[up][i].i)*CarRunTime;
                    charge1=RY(Car[station[up][i].num].R_Y, j-station[up][i].i);
                    if(time1+time2<time) {
                        time=time1+time2;
                        charge=charge1+charge2;
                    }

                    Get1(station[up][i].num, choose, 'F', fb, Car[station[up][i].num].forward[j]);
                }
                else if(time1+time2==time&&charge1+charge2<charge) {
                    charge=charge1+charge2;
                }

                Get1(station[up][i].num, choose, 'F', fb, Car[station[up][i].num].forward[j]);
            }
        }
    }
}

```

```

    }
    else if(station[up][i].F_B=='B'){
        for(j=station[up][i].i+1; j<Car[station[up][i].num].backward.size(); j++){
            charge2=time2=Max;

            if(Change0(Car[station[up][i].num].backward[j], down, choose, charge2, time2, fb)){
                time1=(j-station[up][i].i)*CarRunTime;
                charge1=RY(Car[station[up][i].num].R_Y, j-station[up][i].i);
                if(time1+time2<time){
                    time=time1+time2;
                    charge=charge1+charge2;

                    Get1(station[up][i].num, choose, 'B', fb, Car[station[up][i].num].backward[j]);
                }
                else if(time1+time2==time&&charge1+charge2<charge){
                    charge=charge1+charge2;

                    Get1(station[up][i].num, choose, 'B', fb, Car[station[up][i].num].backward[j]);
                }
            }
        }
    }
}

if(time!=Max){
    outfile<<"从S"<<up<<"到S"<<down<<"，你可以先乘坐L"<<car1<<"在S"<<at<<"换乘
L"<<car2<<endl;
    outfile<<"所花时间为："<<time+CarCarTime<<ends<<"所需费用为："<<charge<<endl;
    outfile<<"L"<<car1<<"行车路线为：";
    Output0(up, at, car1, fb1);
    outfile<<endl<<"L"<<car2<<"行车路线为：";
    Output0(at, down, car2, fb2);
    outfile<<endl<<endl;
}
else
    outfile<<"从S"<<up<<"到S"<<down<<"，你不能通过换乘一次车来达到目的地
"<<endl<<endl;
}

void Change2()
{
    int i, j, k, h, choose, charge=Max, time=Max;
    char fb;
    for(i=0; i<station[up].size(); i++){
        if(station[up][i].F_B=='F'){
            for(j=station[up][i].i+1; j<Car[station[up][i].num].forward.size(); j++){
                for(k=0; k<station[down].size(); k++){
                    if(station[down][k].F_B=='F'){
                        for(h=station[down][k].i-1; h>=0; h--){
                            charge2=time2=Max;

                            if(Change0(Car[station[up][i].num].forward[j], Car[station[down][k].num].forward[h],
                                choose, charge2, time2, fb)){

```



```

        timel=(j-station[up][i].i)*CarRunTime;
        charge1=RY(Car[station[up][i].num].R_Y, j-
station[up][i].i);
        time3=(station[down][k].i-h)*CarRunTime;

        charge3=RY(Car[station[down][k].num].R_Y, station[down][k].i-h);
        if(time1+time2+time3<time) {
            time=time1+time2+time3;
            charge=charge1+charge2+charge3;

            Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'F', Car[station[up][i].
num].forward[j], Car[station[down][k].num].forward[h]);
        }
        else
if(time1+time2+time3==time&&charge1+charge2+charge3<charge) {
            charge=charge1+charge2+charge3;

            Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'F', Car[station[up][i].
num].forward[j], Car[station[down][k].num].forward[h]);
        }
    }
}
else if(station[down][k].F_B=='B') {
    for(h=station[down][k].i-1; h>=0; h--) {
        charge2=time2=Max;

        if(Change0(Car[station[up][i].num].forward[j], Car[station[down][k].num].backward[h]
, choose, charge2, time2, fb)) {
            timel=(j-station[up][i].i)*CarRunTime;
            charge1=RY(Car[station[up][i].num].R_Y, j-
station[up][i].i);
            time3=(station[down][k].i-h)*CarRunTime;

            charge3=RY(Car[station[down][k].num].R_Y, station[down][k].i-h);
            if(time1+time2+time3<time) {
                time=time1+time2+time3;
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'B', Car[station[up][i].
num].forward[j], Car[station[down][k].num].backward[h]);
            }
            else
if(time1+time2+time3==time&&charge1+charge2+charge3<charge) {
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'B', Car[station[up][i].
num].forward[j], Car[station[down][k].num].backward[h]);
            }
        }
    }
}

```

```

    }
    }
}
else if(station[up][i].F_B=='B'){
    for(j=station[up][i].i+1; j<Car[station[up][i].num].backward.size(); j++){
        for(k=0; k<station[down].size(); k++){
            if(station[down][k].F_B=='F'){
                for(h=station[down][k].i-1; h>=0; h--){
                    charge2=time2=Max;

                    if(Change0(Car[station[up][i].num].backward[j], Car[station[down][k].num].forward[h]
, choose, charge2, time2, fb)){
                        time1=(j-station[up][i].i)*CarRunTime;
                        charge1=RY(Car[station[up][i].num].R_Y, j-
station[up][i].i);
                        time3=(station[down][k].i-h)*CarRunTime;

                        charge3=RY(Car[station[down][k].num].R_Y, station[down][k].i-h);
                        if(time1+time2+time3<time){
                            time=time1+time2+time3;
                            charge=charge1+charge2+charge3;

                            Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'F', Car[station[up][i].
num].backward[j], Car[station[down][k].num].forward[h]);
                        }
                        else
if(time1+time2+time3==time&&charge1+charge2+charge3<charge){
                            charge=charge1+charge2+charge3;

                            Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'F', Car[station[up][i].
num].backward[j], Car[station[down][k].num].forward[h]);
                        }
                    }
                }
            }
        }
    }
    else if(station[down][k].F_B=='B'){
        for(h=station[down][k].i-1; h>=0; h--){
            charge2=time2=Max;

            if(Change0(Car[station[up][i].num].backward[j], Car[station[down][k].num].backward[
h], choose, charge2, time2, fb)){
                time1=(j-station[up][i].i)*CarRunTime;
                charge1=RY(Car[station[up][i].num].R_Y, j-
station[up][i].i);
                time3=(station[down][k].i-h)*CarRunTime;

                charge3=RY(Car[station[down][k].num].R_Y, station[down][k].i-h);
                if(time1+time2+time3<time){
                    time=time1+time2+time3;
                    charge=charge1+charge2+charge3;

```



```

        outfile<<"从S"<<up<<"到S"<<down<<"无直达车"<<endl<<endl;
    }
    Change1();
    Change2();
    return 0;
}

```

### 附录3

```

#include <iostream>
#include <fstream>
#include <vector>
#include <iomanip>
#include <string>
using namespace std;

ifstream infile("data.txt", ifstream::in);
ifstream infile1("data1.txt", ifstream::in);
ifstream infile2("data2.txt", ifstream::in);
ifstream infile3("data3.txt", ifstream::in);
ofstream outfile("result.txt", ofstream::out);

typedef struct
{
    char F_B;
    int i;
    int num;
    char C_R;
} tool;

vector< vector<tool> > station(4000);

struct node
{
    vector<int> forward;
    vector<int> backward;
    char R_Y;
    char X_FB_C;
    char C_R;
} Tool[523];

vector< vector<int> > Link(40);

const int Max=1000;
const int CarNum=520;
const int StationNum=3957;
const int CarRunTime=3;
const double RailRunTime=2.5;
const int CarCarTime=5;
const int RailRailTime=4;
const int CarRailTime=7;
const int RailCarTime=6;

```

```

int up, down;
int tool1, tool2, tool3;
int charge1, charge2, charge3;
char fb1, fb2, fb3;
char cr1, cr2, cr3;
double time1, time2, time3;
int at, at1, at2;

void Data_Input()
{
    char l, ch;
    int car_i, zd, i, j;
    tool c;
    infile>>l;
    while(infile>>car_i) {
        infile>>Tool[car_i].R_Y;
        infile>>Tool[car_i].X_FB_C;
        Tool[car_i].C_R='C';
        if(Tool[car_i].X_FB_C=='X') {
            i=0;
            while(infile>>ch&&ch=='S') {
                infile>>zd;
                Tool[car_i].forward.push_back(zd);
                c.i=i++;
                c.num=car_i;
                c.F_B='F';
                c.C_R='C';
                station[zd].push_back(c);
            }
            for(i--, j=0; i>=0; i--, j++) {
                zd=Tool[car_i].forward[i];
                Tool[car_i].backward.push_back(zd);
                c.num=car_i;
                c.i=j;
                c.F_B='B';
                c.C_R='C';
                station[zd].push_back(c);
            }
        }
        else if(Tool[car_i].X_FB_C=='F') {
            i=0;
            while(infile>>ch&&ch=='S') {
                infile>>zd;
                Tool[car_i].forward.push_back(zd);
                c.i=i++;
                c.num=car_i;
                c.F_B='F';
                c.C_R='C';
                station[zd].push_back(c);
            }
        }
    }
}

```

```

        i=0;
        while(infile>>ch&&ch=='S') {
            infile>>zd;
            Tool[car_i].backward.push_back(zd);
            c.i=i++;
            c.num=car_i;
            c.F_B='B';
            c.C_R='C';
            station[zd].push_back(c);
        }
    }
    else if(Tool[car_i].X_FB_C=='C') {
        i=0;
        while(infile>>ch&&ch=='S') {
            infile>>zd;
            Tool[car_i].forward.push_back(zd);
            c.i=i++;
            c.num=car_i;
            c.F_B='F';
            c.C_R='C';
            station[zd].push_back(c);
        }
    }
    if(ch=='E') {
        break;
    }
}
}

```

```

void Data1_Input()
{
    char ch;
    int rail_i, zd, zd1, i, j;
    tool r;
    vector<tool> t;
    infile1>>ch;
    while(ch!='E') {
        infile1>>rail_i;
        rail_i+=CarNum;
        infile1>>Tool[rail_i].X_FB_C;
        Tool[rail_i].C_R='R';
        if(Tool[rail_i].X_FB_C=='X') {
            i=0;
            while(infile1>>ch&&ch=='D') {
                infile1>>zd;
                zd+=StationNum;
                Tool[rail_i].forward.push_back(zd);
                r.i=i++;
                r.num=rail_i;
                r.F_B='F';
                r.C_R='R';
            }
        }
    }
}

```

```

        station[zd].push_back(r);
    }
    for(i--, j=0; i>=0; i--, j++) {
        zd=Tool[rail_i].forward[i];
        Tool[rail_i].backward.push_back(zd);
        r.num=rail_i;
        r.i=j;
        r.F_B='B';
        r.C_R='R';
        station[zd].push_back(r);
    }
}
else if(Tool[rail_i].X_FB_C=='C') {
    i=0;
    while(infile1>>ch&&ch=='D') {
        infile1>>zd;
        zd+=StationNum;
        Tool[rail_i].forward.push_back(zd);
        r.i=i++;
        r.num=rail_i;
        r.F_B='F';
        r.C_R='R';
        station[zd].push_back(r);
    }
}
}
infile2>>ch;
while(ch!='E') {
    infile2>>zd;
    t.clear();
    while(infile2>>ch&&ch=='S') {
        infile2>>zd1;
        Link[zd].push_back(zd1);
        for(i=0; i<station[zd1].size(); i++)
            t.push_back(station[zd1][i]);
    }
    for(i=0; i<station[zd+StationNum].size(); i++)
        t.push_back(station[zd+StationNum][i]);
    station[zd+StationNum]=t;
    for(i=0; i<Link[zd].size(); i++)
        station[Link[zd][i]]=t;
}
infile3>>ch;
while(ch!='E') {
    infile3>>zd;
    t.clear();
    while(infile3>>ch&&ch=='S') {
        infile3>>zd1;
        Link[zd].push_back(zd1);
        for(i=0; i<station[zd1].size(); i++)
            t.push_back(station[zd1][i]);
    }
}

```

```

    }
    for(i=0; i<station[zd+StationNum].size(); i++)
        t.push_back(station[zd+StationNum][i]);
    station[zd+StationNum]=t;
    for(i=0; i<Link[zd].size(); i++)
        station[Link[zd][i]]=t;
}
}

bool Go_Input()
{
    string u,d;
    cout<<"请输入起始站点和目的站点:";
    cin>>u>>d;
    up=1000*(u[1]-'0')+100*(u[2]-'0')+10*(u[3]-'0')+(u[4]-'0');
    down=1000*(d[1]-'0')+100*(d[2]-'0')+10*(d[3]-'0')+(d[4]-'0');
    return false;
}

void Output0(int up,int down,int choose,char fb,char cr)
{
    int i,j,k;
    for(i=0; i<station[up].size(); i++)
        if(station[up][i].num==choose&&station[up][i].F_B==fb)
            break;
    for(j=0; j<station[down].size(); j++)
        if(station[down][j].num==choose&&station[down][j].F_B==fb)
            break;
    if(cr=='C'){
        if(fb=='F'){
            for(k=station[up][i].i; k<=station[down][j].i-1; k++)
                outfile<<"S"<<Tool[station[up][i].num].forward[k]<<"-";
            outfile<<"S"<<Tool[station[up][i].num].forward[k];
        }
        else if(fb=='B'){
            for(k=station[up][i].i; k<=station[down][j].i-1; k++)
                outfile<<"S"<<Tool[station[up][i].num].backward[k]<<"-";
            outfile<<"S"<<Tool[station[up][i].num].backward[k];
        }
    }
    else{
        if(fb=='F'){
            for(k=station[up][i].i; k<=station[down][j].i-1; k++)
                outfile<<"D"<<Tool[station[up][i].num].forward[k]-StationNum<<"-";
            outfile<<"D"<<Tool[station[up][i].num].forward[k]-StationNum;
        }
        else if(fb=='B'){
            for(k=station[up][i].i; k<=station[down][j].i-1; k++)
                outfile<<"D"<<Tool[station[up][i].num].backward[k]-StationNum<<"-";
            outfile<<"D"<<Tool[station[up][i].num].backward[k]-StationNum;
        }
    }
}

```



```

    }
}

int RY(char ry, int t)
{
    if(ry=='R') {
        if(t<=20)
            return 1;
        else if(t<=40)
            return 2;
        else
            return 3;
    }
    return 1;
}

void Get1(int t1, int t2, char f1, char f2, char r1, char r2, int a) {
    tool1=t1;tool2=t2;
    fb1=f1;fb2=f2;
    cr1=r1,cr2=r2;
    at=a;
}

void Get2(int t1, int t2, int t3, char f1, char f2, char f3, char r1, char r2, char r3, int
a1, int a2)
{
    tool1=t1;tool2=t2;tool3=t3;
    fb1=f1;fb2=f2;fb3=f3;
    cr1=r1,cr2=r2;cr3=r3;
    at1=a1;at2=a2;
}

bool Change0(int up, int down, int &choose, int &charge, double &time, char &fb, char &cr)
{
    int i, j, t;
    for(i=0; i<station[up].size(); i++) {
        for(j=0; j<station[down].size(); j++) {

            if(station[up][i].num==station[down][j].num&&station[up][i].F_B==station[down][j].
F_B&&station[up][i].i<station[down][j].i) {
                t=station[down][j].i-station[up][i].i;
                if(station[up][i].C_R=='C') {
                    if(Tool[station[up][i].num].R_Y=='R') {
                        if(t<=20) {
                            if(charge>1) {
                                charge=1;
                                time=t*CarRunTime;
                                choose=station[up][i].num;
                                fb=station[up][i].F_B;
                                cr='C';
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        else if(charge==1&& t*CarRunTime<time) {
            time=t*CarRunTime;
            choose=station[up][i].num;
            fb=station[up][i].F_B;
            cr='C' ;
        }
    }
    else if(t<=40) {
        if(charge>2) {
            charge=2;
            time=t*CarRunTime;
            choose=station[up][i].num;
            fb=station[up][i].F_B;
            cr='C' ;
        }
        else if(charge==2&& t*CarRunTime<time) {
            time=t*CarRunTime;
            choose=station[up][i].num;
            fb=station[up][i].F_B;
            cr='C' ;
        }
    }
    else{
        if(charge>3) {
            charge=3;
            time=t*CarRunTime;
            choose=station[up][i].num;
            fb=station[up][i].F_B;
            cr='C' ;
        }
        else if(charge==3&& t*CarRunTime<time) {
            time=t*CarRunTime;
            choose=station[up][i].num;
            fb=station[up][i].F_B;
            cr='C' ;
        }
    }
}
else{
    if(charge>1) {
        charge=1;
        time=t*CarRunTime;
        choose=station[up][i].num;
        fb=station[up][i].F_B;
        cr='C' ;
    }
    else if(charge==1&& t*CarRunTime<time) {
        time=t*CarRunTime;
        choose=station[up][i].num;
        fb=station[up][i].F_B;
        cr='C' ;
    }
}

```

```

        }
    }
}
else{
    if(charge>3){
        charge=3;
        time=t*RailRunTime;
        choose=station[up][i].num;
        fb=station[up][i].F_B;
        cr='R';
    }
    else if(charge==3&& t*RailRunTime<time){
        time=t*RailRunTime;
        choose=station[up][i].num;
        fb=station[up][i].F_B;
        cr='R';
    }
}
}
}
}
if(charge==Max)
    return false;
return true;
}

void Change1()
{
    int i, j, choose, charge=Max;
    char fb, cr;
    double time=Max;
    for(i=0; i<station[up].size(); i++){
        if(station[up][i].F_B=='F'){
            for(j=station[up][i].i+1; j<Tool[station[up][i].num].forward.size(); j++){
                time2=charge2=Max;

                if(Change0(Tool[station[up][i].num].forward[j], down, choose, charge2, time2, fb, cr)){
                    time1=(j-station[up][i].i)*CarRunTime;
                    if(station[up][i].C_R=='C'){
                        charge1=RY(Tool[station[up][i].num].R_Y, j-station[up][i].i);
                        if(charge1+charge2<charge){
                            charge=charge1+charge2;
                            if(cr=='C')
                                time=time1+time2+CarCarTime;
                            else
                                time=time1+time2+CarRailTime;
                        }
                    }
                    Get1(station[up][i].num, choose, 'F', fb, 'C', cr, Tool[station[up][i].num].forward[j]);
                }
                else if(charge1+charge2==charge){
                    if(cr=='C' && time>time1+time2+CarCarTime){

```

```

        time=time1+time2+CarCarTime;

Get1(station[up][i].num,choose,'F',fb,'C',cr,Tool[station[up][i].num].forward[j]);
    }
    else if(cr=='R' &&time>time1+time2+CarRailTime){
        time=time1+time2+CarRailTime;

Get1(station[up][i].num,choose,'F',fb,'C',cr,Tool[station[up][i].num].forward[j]);
    }
}
else{
    if(cr=='C')
        charge1=3;
    else
        charge1=0;
    if(charge1+charge2<charge){
        charge=charge1+charge2;
        if(cr=='C')
            time=time1+time2+RailCarTime;
        else
            time=time1+time2+RailRailTime;

Get1(station[up][i].num,choose,'F',fb,'R',cr,Tool[station[up][i].num].forward[j]);
    }
    else if(charge1+charge2==charge){
        if(cr=='C' &&time>time1+time2+RailCarTime){
            time=time1+time2+RailCarTime;

Get1(station[up][i].num,choose,'F',fb,'R',cr,Tool[station[up][i].num].forward[j]);
        }
        else if(cr=='R' &&time>time1+time2+RailRailTime){
            time=time1+time2+RailRailTime;

Get1(station[up][i].num,choose,'F',fb,'R',cr,Tool[station[up][i].num].forward[j]);
        }
    }
}
}
else if(station[up][i].F_B=='B'){
    for(j=station[up][i].i+1; j<Tool[station[up][i].num].backward.size();
j++) {
        time2=charge2=Max;

if(Change0(Tool[station[up][i].num].backward[j],down,choose,charge2,time2,fb,cr)){
    time1=(j-station[up][i].i)*CarRunTime;
    if(station[up][i].C_R=='C'){
        charge1=RY(Tool[station[up][i].num].R_Y,j-station[up][i].i);
        if(charge1+charge2<charge){

```



```

}
if(charge!=Max){
    if(cr1=='C'){
        outfile<<"从S"<<up<<"到S"<<down<<"，你可以先乘坐L"<<tool1;
    }
    else
        outfile<<"从S"<<up<<"到S"<<down<<"，你可以先乘坐T"<<tool1-CarNum;
    if(at<=StationNum)
        outfile<<"在S"<<at<<"换乘";
    else
        outfile<<"在D"<<at-StationNum<<"换乘";
    if(cr2=='C'){
        outfile<<"L"<<tool2<<endl;
    }
    else
        outfile<<"T"<<tool2-CarNum<<endl;
    outfile<<"所需费用为:"<<charge<<ends<<"所花时间为:"<<time<<endl;
    if(cr1=='C'){
        outfile<<"L"<<tool1<<"行车路线为:";
    }
    else
        outfile<<"D"<<tool1-CarNum<<"行车路线为:";
    Output0(up, at, tool1, fb1, cr1);
    if(cr2=='C'){
        outfile<<endl<<"L"<<tool2<<"行车路线为:";
    }
    else
        outfile<<endl<<"D"<<tool2-CarNum<<"行车路线为:";
    Output0(at, down, tool2, fb2, cr2);
    outfile<<endl<<endl;
}
else
    outfile<<"从S"<<up<<"到S"<<down<<"，你不能通过换乘一次来达到目的地"<<endl<<endl;
}

void Change2()
{
    int i, j, k, h, choose, charge=Max;
    char fb, cr;
    double time=Max;
    for(i=0; i<station[up].size(); i++){
        if(station[up][i].F_B=='F'){
            for(j=station[up][i].i+1; j<Tool[station[up][i].num].forward.size(); j++){
                for(k=0; k<station[down].size(); k++){
                    if(station[down][k].F_B=='F'){
                        for(h=station[down][k].i-1; h>=0; h--){
                            time2=charge2=Max;

                            if(Change0(Tool[station[up][i].num].forward[j], Tool[station[down][k].num].forward[
h], choose, charge2, time2, fb, cr)){
                                timel=(j-station[up][i].i)*CarRunTime;
                                time3=(station[down][k].i-h)*CarRunTime;

                                if(station[up][i].C_R=='C' && station[down][k].C_R=='C'){
                                    charge1=RY(Tool[station[up][i].num].R_Y, j-
station[up][i].i);

```

```

charge3=RY(Tool[station[down][k].num].R_Y,station[down][k].i-h);
        if(charge1+charge2+charge3<charge){
            charge=charge1+charge2+charge3;
            if(cr=='C')
                time=time1+time2+time3+CarCarTime*2;
            else

time=time1+time2+time3+CarRailTime+RailCarTime;

        Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','C',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
        }
        else if(charge1+charge2+charge3==charge){

if(cr=='C' &&charge1+charge2+charge3+CarCarTime*2<time){
            time=time1+time2+time3+CarCarTime*2;

        Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','C',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
        }
        else
if(cr=='R' &&time1+time2+time3+CarRailTime+RailCarTime<time){

time=time1+time2+time3+CarRailTime+RailCarTime;

        Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','C',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
        }
        }
        else
if(station[up][i].C_R=='C' &&station[down][k].C_R=='R'){
            charge1=RY(Tool[station[up][i].num].R_Y,j-
station[up][i].i);

            if(cr=='C')
                charge3=3;
            else
                charge3=0;
            if(charge1+charge2+charge3<charge){
                charge=charge1+charge2+charge3;
                if(cr=='C')

time=time1+time2+time3+CarCarTime+CarRailTime;
                else

time=time1+time2+time3+CarRailTime+RailRailTime;

        Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','C',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
        }

```

```

else if(charge1+charge2+charge3==charge) {

    if(cr=='C' &&time1+time2+time3+CarCarTime+CarRailTime<time) {

        time=time1+time2+time3+CarCarTime+CarRailTime;

        Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','C',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
    }
    else
    if(cr=='R' &&time1+time2+time3+CarRailTime+RailRailTime<time) {

        time=time1+time2+time3+CarRailTime+RailRailTime;

        Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','C',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
    }
    }
    else
    if(station[up][i].C_R=='R' &&station[down][k].C_R=='C') {
        if(cr=='C')
            charge1=3;
        else
            charge1=0;

        charge3=RY(Tool[station[down][k].num].R_Y,station[down][k].i-h);
        if(charge1+charge2+charge3<charge) {
            charge=charge1+charge2+charge3;
            if(cr=='C')

                time=time1+time2+time3+RailCarTime+CarCarTime;
            else

                time=time1+time2+time3+RailRailTime+RailCarTime;

            Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','R',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
        }
        else if(charge1+charge2+charge3==charge) {

            if(cr=='C' &&time1+time2+time3+RailCarTime+CarCarTime<time) {

                time=time1+time2+time3+RailCarTime+CarCarTime;

                Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','R',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
            }
            else
            if(cr=='R' &&time1+time2+time3+RailRailTime+RailCarTime<time) {

```





```

        if(Change0(Tool[station[up][i].num].forward[j],Tool[station[down][k].num].backward
[h],choose,charge2,time2,fb,cr)){
            time1=(j-station[up][i].i)*CarRunTime;
            time3=(station[down][k].i-h)*CarRunTime;

            if(station[up][i].C_R=='C' &&station[down][k].C_R=='C'){
                charge1=RY(Tool[station[up][i].num].R_Y,j-
station[up][i].i);

                charge3=RY(Tool[station[down][k].num].R_Y,station[down][k].i-h);
                if(charge1+charge2+charge3<charge){
                    charge=charge1+charge2+charge3;
                    if(cr=='C')
                        time=time1+time2+time3+CarCarTime*2;
                    else

time=time1+time2+time3+CarRailTime+RailCarTime;

                Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','C',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
                }
                else if(charge1+charge2+charge3==charge){

                    if(cr=='C' &&charge1+charge2+charge3+CarCarTime*2<time){
                        time=time1+time2+time3+CarCarTime*2;

                        Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','C',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
                        }
                        else
if(cr=='R' &&time1+time2+time3+CarRailTime+RailCarTime<time){

                            time=time1+time2+time3+CarRailTime+RailCarTime;

                            Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','C',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
                            }
                        }
                    }
                else
if(station[up][i].C_R=='C' &&station[down][k].C_R=='R'){
                    charge1=RY(Tool[station[up][i].num].R_Y,j-
station[up][i].i);

                    if(cr=='C')
                        charge3=3;
                    else
                        charge3=0;
                    if(charge1+charge2+charge3<charge){
                        charge=charge1+charge2+charge3;
                        if(cr=='C')

```

```

time=time1+time2+time3+CarCarTime+CarRailTime;
    else

time=time1+time2+time3+CarRailTime+RailRailTime;

    Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','C',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
    }
    else if(charge1+charge2+charge3==charge){

if(cr=='C' &&time1+time2+time3+CarCarTime+CarRailTime<time){

time=time1+time2+time3+CarCarTime+CarRailTime;

    Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','C',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
    }
    else
if(cr=='R' &&time1+time2+time3+CarRailTime+RailRailTime<time){

time=time1+time2+time3+CarRailTime+RailRailTime;

    Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','C',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
    }
    }
    else
if(station[up][i].C_R=='R' &&station[down][k].C_R=='C'){
    if(cr=='C')
        charge1=3;
    else
        charge1=0;

charge3=RY(Tool[station[down][k].num].R_Y,station[down][k].i-h);
    if(charge1+charge2+charge3<charge){
        charge=charge1+charge2+charge3;
        if(cr=='C')

time=time1+time2+time3+RailCarTime+CarCarTime;
    else

time=time1+time2+time3+RailRailTime+RailCarTime;

    Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','R',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
    }
    else if(charge1+charge2+charge3==charge){

if(cr=='C' &&time1+time2+time3+RailCarTime+CarCarTime<time){

```

```

time=time1+time2+time3+RailCarTime+CarCarTime;

Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','R',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
    }
    else
if(cr=='R' &&time1+time2+time3+RailRailTime+RailCarTime<time) {

time=time1+time2+time3+RailRailTime+RailCarTime;

Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','R',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
    }
}
else{
    if(cr=='C')
        charge1=3;
    else
        charge1=0;
    if(cr=='C')
        charge3=3;
    else
        charge3=0;
    if(charge1+charge2+charge3<charge) {
        charge=charge1+charge2+charge3;
        if(cr=='C')

time=time1+time2+time3+RailCarTime+CarRailTime;
        else
            time=time1+time2+time3+RailRailTime*2;

Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','R',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
    }
    else if(charge1+charge2+charge3==charge) {

if(cr=='C' &&time1+time2+time3+RailCarTime+CarRailTime<time) {

time=time1+time2+time3+RailCarTime+CarRailTime;

Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','R',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
    }
    else
if(cr=='R' &&time1+time2+time3+RailRailTime*2<time) {
        time=time1+time2+time3+RailRailTime*2;

Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','R',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);

```



```

        Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'F', 'C', cr, 'C', Tool[sta
tion[up][i].num].backward[j], Tool[station[down][k].num].forward[h]);
    }
}
else
if(station[up][i].C_R=='C' &&station[down][k].C_R=='R') {
    charge1=RY(Tool[station[up][i].num].R_Y, j-
station[up][i].i);
    if(cr=='C')
        charge3=3;
    else
        charge3=0;
    if(charge1+charge2+charge3<charge) {
        charge=charge1+charge2+charge3;
        if(cr=='C')

time=time1+time2+time3+CarCarTime+CarRailTime;
        else

time=time1+time2+time3+CarRailTime+RailRailTime;

        Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'F', 'C', cr, 'R', Tool[sta
tion[up][i].num].backward[j], Tool[station[down][k].num].forward[h]);
    }
    else if(charge1+charge2+charge3==charge) {

if(cr=='C' &&time1+time2+time3+CarCarTime+CarRailTime<time) {

time=time1+time2+time3+CarCarTime+CarRailTime;

        Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'F', 'C', cr, 'R', Tool[sta
tion[up][i].num].backward[j], Tool[station[down][k].num].forward[h]);
    }
    else
if(cr=='R' &&time1+time2+time3+CarRailTime+RailRailTime<time) {

time=time1+time2+time3+CarRailTime+RailRailTime;

        Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'F', 'C', cr, 'R', Tool[sta
tion[up][i].num].backward[j], Tool[station[down][k].num].forward[h]);
    }
}
}
else
if(station[up][i].C_R=='R' &&station[down][k].C_R=='C') {
    if(cr=='C')
        charge1=3;
    else
        charge1=0;

```

```

charge3=RY(Tool[station[down][k].num].R_Y,station[down][k].i-h);
        if(charge1+charge2+charge3<charge){
            charge=charge1+charge2+charge3;
            if(cr=='C')

time=time1+time2+time3+RailCarTime+CarCarTime;
            else

time=time1+time2+time3+RailRailTime+RailCarTime;

        Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','R',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);
        }
        else if(charge1+charge2+charge3==charge){

if(cr=='C' &&time1+time2+time3+RailCarTime+CarCarTime<time){

time=time1+time2+time3+RailCarTime+CarCarTime;

        Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','R',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);
        }
        else
if(cr=='R' &&time1+time2+time3+RailRailTime+RailCarTime<time){

time=time1+time2+time3+RailRailTime+RailCarTime;

        Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','R',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);
        }
    }
}
else{
    if(cr=='C')
        charge1=3;
    else
        charge1=0;
    if(cr=='C')
        charge3=3;
    else
        charge3=0;
    if(charge1+charge2+charge3<charge){
        charge=charge1+charge2+charge3;
        if(cr=='C')

time=time1+time2+time3+RailCarTime+CarRailTime;
        else
            time=time1+time2+time3+RailRailTime*2;

        Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','R',cr,'R',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);

```

```

    }
    else if(charge1+charge2+charge3==charge) {

        if(cr=='C' &&time1+time2+time3+RailCarTime+CarRailTime<time) {

            time=time1+time2+time3+RailCarTime+CarRailTime;

            Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','R',cr,'R',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);
        }
        else
        if(cr=='R' &&time1+time2+time3+RailRailTime*2<time) {
            time=time1+time2+time3+RailRailTime*2;

            Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','R',cr,'R',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);
        }
    }
}
}
}
}
}
else if(station[down][k].F_B=='B') {
    for(h=station[down][k].i-1; h>=0; h--) {
        time2=charge2=Max;

        if(Change0(Tool[station[up][i].num].backward[j],Tool[station[down][k].num].backwar
d[h],choose,charge2,time2,fb,cr)) {
            time1=(j-station[up][i].i)*CarRunTime;
            time3=(station[down][k].i-h)*CarRunTime;

            if(station[up][i].C_R=='C' &&station[down][k].C_R=='C') {
                charge1=RY(Tool[station[up][i].num].R_Y,j-
station[up][i].i);

                charge3=RY(Tool[station[down][k].num].R_Y,station[down][k].i-h);
                if(charge1+charge2+charge3<charge) {
                    charge=charge1+charge2+charge3;
                    if(cr=='C')
                        time=time1+time2+time3+CarCarTime*2;
                    else

time=time1+time2+time3+CarRailTime+RailCarTime;

                Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','C',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
            }
            else if(charge1+charge2+charge3==charge) {

                if(cr=='C' &&time1+time2+time3+CarCarTime*2<time) {
                    time=time1+time2+time3+CarCarTime*2;

```



```

        Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','C',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
    }
    else
if(cr=='R' &&time1+time2+time3+CarRailTime+RailCarTime<time) {

    time=time1+time2+time3+CarRailTime+RailCarTime;

    Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','C',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
    }
    }
    else
if(station[up][i].C_R=='C' &&station[down][k].C_R=='R') {
    charge1=RY(Tool[station[up][i].num].R_Y,j-
station[up][i].i);

    if(cr=='C')
        charge3=3;
    else
        charge3=0;
    if(charge1+charge2+charge3<charge) {
        charge=charge1+charge2+charge3;
        if(cr=='C')

time=time1+time2+time3+CarCarTime+CarRailTime;
        else

time=time1+time2+time3+CarRailTime+RailRailTime;

        Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','C',cr,'R',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
    }
    else if(charge1+charge2+charge3==charge) {

if(cr=='C' &&time1+time2+time3+CarCarTime+CarRailTime<time) {

time=time1+time2+time3+CarCarTime+CarRailTime;

        Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','C',cr,'R',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
    }
    else
if(cr=='R' &&time1+time2+time3+CarRailTime+RailRailTime<time) {

time=time1+time2+time3+CarRailTime+RailRailTime;

        Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','C',cr,'R',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
    }
}

```

```

        }
    }
    else
    if(station[up][i].C_R=='R' &&station[down][k].C_R=='C') {
        if(cr=='C')
            charge1=3;
        else
            charge1=0;

        charge3=RY(Tool[station[down][k].num].R_Y,station[down][k].i-h);
        if(charge1+charge2+charge3<charge) {
            charge=charge1+charge2+charge3;
            if(cr=='C')

            time=time1+time2+time3+RailCarTime+CarCarTime;
            else

            time=time1+time2+time3+RailRailTime+RailCarTime;

            Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','R',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
        }
        else if(charge1+charge2+charge3==charge) {

        if(cr=='C' &&time1+time2+time3+RailCarTime+CarCarTime<time) {

            time=time1+time2+time3+RailCarTime+CarCarTime;

            Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','R',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
        }
        else
        if(cr=='R' &&time1+time2+time3+RailRailTime+RailCarTime<time) {

            time=time1+time2+time3+RailRailTime+RailCarTime;

            Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','R',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
        }
    }
}
else{
    if(cr=='C')
        charge1=3;
    else
        charge1=0;
    if(cr=='C')
        charge3=3;
    else
        charge3=0;
    if(charge1+charge2+charge3<charge) {

```



```

        outfile<<"再在D"<<at2-StationNum<<"换乘";
    if(cr3=='C')
        outfile<<"L"<<tool3<<endl;
    else
        outfile<<'T'<<tool3-CarNum<<endl;
    outfile<<"所需费用为:"<<charge<<ends<<"所花时间为:"<<time<<endl;
    if(cr1=='C')
        outfile<<"L"<<tool1<<"行车路线为:";
    else
        outfile<<"T"<<tool1-CarNum<<"行车路线为:";
    Output0(up, at1, tool1, fb1, cr1);
    if(cr2=='C')
        outfile<<endl<<"L"<<tool2<<"行车路线为:";
    else
        outfile<<endl<<"T"<<tool2-CarNum<<"行车路线为:";
    Output0(at1, at2, tool2, fb2, cr2);
    if(cr3=='C')
        outfile<<endl<<"L"<<tool3<<"行车路线为:";
    else
        outfile<<endl<<"T"<<tool3-CarNum<<"行车路线为:";
    Output0(at2, down, tool3, fb3, cr3);
    outfile<<endl<<endl;
} else
    outfile<<"从S"<<up<<"到S"<<down<<"，你不能通过换乘二次来达到目的地"<<endl<<endl;
}

int main()
{
    Data_Input();
    Data1_Input();
    Go_Input();
    int choose;
    int charge=Max;
    char fb, cr;
    double time=Max;
    if(Change0(up, down, choose, charge, time, fb, cr)){
        if(cr=='C')
            outfile<<"从S"<<up<<"到S"<<down<<"，你可以乘坐直达车L"<<choose<<endl;
        else
            outfile<<"从S"<<up<<"到S"<<down<<"，你可以乘坐地铁T"<<choose-CarNum<<endl;
        outfile<<"所需费用为:"<<charge<<ends<<"所花时间为:"<<time<<endl;
        outfile<<"行车路线为: ";
        Output0(up, down, choose, fb, cr);
        outfile<<endl<<endl;
    }
    else{
        outfile<<"从S"<<up<<"到S"<<down<<"无直达车和地铁"<<endl<<endl;
    }
    Change1();
    Change2();
    return 0;
}

```

```
}
```

#### 附录4

```
#include <iostream>
#include <fstream>
#include <vector>
#include <iomanip>
#include <string>
using namespace std;

ifstream infile("data.txt", ifstream::in);
ifstream infile1("data1.txt", ifstream::in);
ifstream infile2("data2.txt", ifstream::in);
ifstream infile3("data3.txt", ifstream::in);
ofstream outfile("result.txt", ofstream::out);

typedef struct
{
    char F_B;
    int i;
    int num;
    char C_R;
}tool;

vector< vector<tool> > station(4000);

struct node
{
    vector<int> forward;
    vector<int> backward;
    char R_Y;
    char X_FB_C;
    char C_R;
}Tool[523];

vector< vector<int> > Link(40);

const int Max=1000;
const int CarNum=520;
const int StationNum=3957;
const int CarRunTime=3;
const double RailRunTime=2.5;
const int CarCarTime=5;
const int RailRailTime=4;
const int CarRailTime=7;
const int RailCarTime=6;

int up, down;
int tool1, tool2, tool3;
int charge1, charge2, charge3;
char fb1, fb2, fb3;
```

```

char cr1, cr2, cr3;
double time1, time2, time3;
int at, at1, at2;

void Data_Input()
{
    char l, ch;
    int car_i, zd, i, j;
    tool c;
    infile>>l;
    while(infile>>car_i) {
        infile>>Tool[car_i].R_Y;
        infile>>Tool[car_i].X_FB_C;
        Tool[car_i].C_R='C';
        if(Tool[car_i].X_FB_C=='X') {
            i=0;
            while(infile>>ch&&ch=='S') {
                infile>>zd;
                Tool[car_i].forward.push_back(zd);
                c.i=i++;
                c.num=car_i;
                c.F_B='F';
                c.C_R='C';
                station[zd].push_back(c);
            }
            for(i--, j=0; i>=0; i--, j++) {
                zd=Tool[car_i].forward[i];
                Tool[car_i].backward.push_back(zd);
                c.num=car_i;
                c.i=j;
                c.F_B='B';
                c.C_R='C';
                station[zd].push_back(c);
            }
        }
        else if(Tool[car_i].X_FB_C=='F') {
            i=0;
            while(infile>>ch&&ch=='S') {
                infile>>zd;
                Tool[car_i].forward.push_back(zd);
                c.i=i++;
                c.num=car_i;
                c.F_B='F';
                c.C_R='C';
                station[zd].push_back(c);
            }
            i=0;
            while(infile>>ch&&ch=='S') {
                infile>>zd;
                Tool[car_i].backward.push_back(zd);
            }
        }
    }
}

```

```

        c.i=i++;
        c.num=car_i;
        c.F_B='B';
        c.C_R='C';
        station[zd].push_back(c);
    }
}
else if(Tool[car_i].X_FB_C=='C'){
    i=0;
    while(infile>>ch&&ch=='S'){
        infile>>zd;
        Tool[car_i].forward.push_back(zd);
        c.i=i++;
        c.num=car_i;
        c.F_B='F';
        c.C_R='C';
        station[zd].push_back(c);
    }
}
if(ch=='E'){
    break;
}
}
}

```

```

void Data1_Input()
{
    char ch;
    int rail_i, zd, zd1, i, j;
    tool r;
    vector<tool> t;
    infile1>>ch;
    while(ch!='E'){
        infile1>>rail_i;
        rail_i+=CarNum;
        infile1>>Tool[rail_i].X_FB_C;
        Tool[rail_i].C_R='R';
        if(Tool[rail_i].X_FB_C=='X'){
            i=0;
            while(infile1>>ch&&ch=='D'){
                infile1>>zd;
                zd+=StationNum;
                Tool[rail_i].forward.push_back(zd);
                r.i=i++;
                r.num=rail_i;
                r.F_B='F';
                r.C_R='R';
                station[zd].push_back(r);
            }
            for(i--, j=0; i>=0; i--, j++){
                zd=Tool[rail_i].forward[i];
            }
        }
    }
}

```

```

        Tool[rail_i].backward.push_back(zd);
        r.num=rail_i;
        r.i=j;
        r.F_B='B';
        r.C_R='R';
        station[zd].push_back(r);
    }
}
else if(Tool[rail_i].X_FB_C=='C'){
    i=0;
    while(infile1>>ch&&ch=='D'){
        infile1>>zd;
        zd+=StationNum;
        Tool[rail_i].forward.push_back(zd);
        r.i=i++;
        r.num=rail_i;
        r.F_B='F';
        r.C_R='R';
        station[zd].push_back(r);
    }
}
}
infile2>>ch;
while(ch!='E'){
    infile2>>zd;
    t.clear();
    while(infile2>>ch&&ch=='S'){
        infile2>>zd1;
        Link[zd].push_back(zd1);
        for(i=0; i<station[zd1].size(); i++)
            t.push_back(station[zd1][i]);
    }
    for(i=0; i<station[zd+StationNum].size(); i++)
        t.push_back(station[zd+StationNum][i]);
    station[zd+StationNum]=t;
    for(i=0; i<Link[zd].size(); i++)
        station[Link[zd][i]]=t;
}
infile3>>ch;
while(ch!='E'){
    infile3>>zd;
    t.clear();
    while(infile3>>ch&&ch=='S'){
        infile3>>zd1;
        Link[zd].push_back(zd1);
        for(i=0; i<station[zd1].size(); i++)
            t.push_back(station[zd1][i]);
    }
    for(i=0; i<station[zd+StationNum].size(); i++)
        t.push_back(station[zd+StationNum][i]);
    station[zd+StationNum]=t;
}

```



```

        for(i=0; i<Link[zd].size(); i++)
            station[Link[zd][i]]=t;
    }
}

bool Go_Input()
{
    string u,d;
    cout<<"请输入起始站点和目的站点:";
    cin>>u>>d;
    up=1000*(u[1]-'0')+100*(u[2]-'0')+10*(u[3]-'0')+(u[4]-'0');
    down=1000*(d[1]-'0')+100*(d[2]-'0')+10*(d[3]-'0')+(d[4]-'0');
    return false;
}

void Output0(int up,int down,int choose_tool,char fb,char cr)
{
    int i,j,k;
    for(i=0; i<station[up].size(); i++)
        if(station[up][i].num==choose_tool&&station[up][i].F_B==fb)
            break;
    for(j=0; j<station[down].size(); j++)
        if(station[down][j].num==choose_tool&&station[down][j].F_B==fb)
            break;
    if(cr=='C') {
        if(fb=='F') {
            for(k=station[up][i].i; k<=station[down][j].i-1; k++)
                outfile<<"S"<<Tool[station[up][i].num].forward[k]<<"->";
            outfile<<"S"<<Tool[station[up][i].num].forward[k];
        }
        else if(fb=='B') {
            for(k=station[up][i].i; k<=station[down][j].i-1; k++)
                outfile<<"S"<<Tool[station[up][i].num].backward[k]<<"->";
            outfile<<"S"<<Tool[station[up][i].num].backward[k];
        }
    }
    else{
        if(fb=='F') {
            for(k=station[up][i].i; k<=station[down][j].i-1; k++)
                outfile<<"D"<<Tool[station[up][i].num].forward[k]-StationNum<<"->";
            outfile<<"D"<<Tool[station[up][i].num].forward[k]-StationNum;
        }
        else if(fb=='B') {
            for(k=station[up][i].i; k<=station[down][j].i-1; k++)
                outfile<<"D"<<Tool[station[up][i].num].backward[k]-StationNum<<"->";
            outfile<<"D"<<Tool[station[up][i].num].backward[k]-StationNum;
        }
    }
}

int RY(char ry,int t)

```

```

{
    if(ry=='R') {
        if(t<=20)
            return 1;
        else if(t<=40)
            return 2;
        else
            return 3;
    }
    return 1;
}

void Get1(int t1, int t2, char f1, char f2, char r1, char r2, int a) {
    tool1=t1;tool2=t2;
    fb1=f1;fb2=f2;
    cr1=r1,cr2=r2;
    at=a;
}

void Get2(int t1, int t2, int t3, char f1, char f2, char f3, char r1, char r2, char r3, int
a1, int a2)
{
    tool1=t1;tool2=t2;tool3=t3;
    fb1=f1;fb2=f2;fb3=f3;
    cr1=r1,cr2=r2;cr3=r3;
    at1=a1;at2=a2;
}

bool Change0(int up, int down, int &choose_tool, int &charge, double &time, char &fb, char
&cr)
{
    int i, j, t;
    for(i=0; i<station[up].size(); i++) {
        for(j=0; j<station[down].size(); j++) {

            if(station[up][i].num==station[down][j].num&&station[up][i].F_B==station[down][j].
F_B&&station[up][i].i<station[down][j].i) {
                t=station[down][j].i-station[up][i].i;
                if(station[up][i].C_R=='C') {
                    if(t*CarRunTime<time) {
                        time=t*CarRunTime;
                        if(Tool[station[up][i].num].R_Y=='R') {
                            if(t<=20) {
                                charge=1;
                                choose_tool=station[up][i].num;
                                fb=station[up][i].F_B;
                                cr='C';
                            }
                        }
                    }
                }
            }
            else if(t<=40) {
                charge=2;
                choose_tool=station[up][i].num;
            }
        }
    }
}

```

```

        fb=station[up][i].F_B;
        cr='C' ;
    }
    else{
        charge=3;
        choose_tool=station[up][i].num;
        fb=station[up][i].F_B;
        cr='C' ;
    }
}
else{
    charge=1;
    choose_tool=station[up][i].num;
    fb=station[up][i].F_B;
    cr='C' ;
}
}
else if(t*CarRunTime==time) {
    if(Tool[station[up][i].num].R_Y=='R') {
        if(t<=20&&charge>1) {
            charge=1;
            choose_tool=station[up][i].num;
            fb=station[up][i].F_B;
            cr='C' ;
        }
        else if(t<=40&&charge>2) {
            charge=2;
            choose_tool=station[up][i].num;
            fb=station[up][i].F_B;
            cr='C' ;
        }
        else if(charge>3) {
            charge=3;
            choose_tool=station[up][i].num;
            fb=station[up][i].F_B;
            cr='C' ;
        }
    }
    else{
        if(charge>1) {
            charge=1;
            choose_tool=station[up][i].num;
            fb=station[up][i].F_B;
            cr='C' ;
        }
    }
}
}
else{
    if(t*RailRunTime<time) {
        time=t*RailRunTime;

```

```

        charge=3;
        choose_tool=station[up][i].num;
        fb=station[up][i].F_B;
        cr='R';
    }
    else if(t*RailRunTime==time){
        if(charge>3){
            charge=3;
            choose_tool=station[up][i].num;
            fb=station[up][i].F_B;
            cr='R';
        }
    }
}

if(time==Max)
    return false;
return true;
}

void Changel()
{
    int i, j, choose, charge=Max;
    char fb, cr;
    double time=Max;
    for(i=0; i<station[up].size(); i++){
        if(station[up][i].F_B=='F'){
            for(j=station[up][i].i+1; j<Tool[station[up][i].num].forward.size(); j++){
                time2=charge2=Max;

                if(Change0(Tool[station[up][i].num].forward[j], down, choose, charge2, time2, fb, cr)){
                    if(station[up][i].C_R=='C'){
                        time1=(j-station[up][i].i)*CarRunTime;
                        charge1=RY(Tool[station[up][i].num].R_Y, j-station[up][i].i);
                        if(cr=='C'){
                            if(time1+time2+CarCarTime<time){
                                time=time1+time2+CarCarTime;
                                charge=charge1+charge2;

                                Get1(station[up][i].num, choose, 'F', fb, 'C', cr, Tool[station[up][i].num].forward[j]);
                            }
                            else
                                if(time1+time2+CarCarTime==time&&charge1+charge2<charge){
                                    charge=charge1+charge2;

                                    Get1(station[up][i].num, choose, 'F', fb, 'C', cr, Tool[station[up][i].num].forward[j]);
                                }
                        }
                    }
                }
            }
        }
    }
}

```

```

        if(time1+time2+CarRailTime<time){
            time=time1+time2+CarRailTime;
            charge=charge1+charge2;

        Get1(station[up][i].num,choose,'F',fb,'C',cr,Tool[station[up][i].num].forward[j]);
        }
        else
if(time1+time2+CarRailTime==time&&charge1+charge2<charge){
            charge=charge1+charge2;

        Get1(station[up][i].num,choose,'F',fb,'C',cr,Tool[station[up][i].num].forward[j]);
        }
    }
    else if(station[up][i].C_R=='R'){
        time1=(j-station[up][i].i)*RailRunTime;
        if(cr=='C'){
            charge1=3;
            if(time1+time2+RailCarTime<time){
                time=time1+time2+RailCarTime;
                charge=charge1+charge2;

            Get1(station[up][i].num,choose,'F',fb,'R',cr,Tool[station[up][i].num].forward[j]);
            }
            else
if(time1+time2+RailCarTime==time&&charge1+charge2<charge){
                charge=charge1+charge2;

            Get1(station[up][i].num,choose,'F',fb,'R',cr,Tool[station[up][i].num].forward[j]);
            }
        }
        else if(cr=='R'){
            charge1=0;
            if(time1+time2+RailRailTime<time){
                time=time1+time2+RailRailTime;
                charge=charge1+charge2;

            Get1(station[up][i].num,choose,'F',fb,'R',cr,Tool[station[up][i].num].forward[j]);
            }
            else
if(time1+time2+RailRailTime==time&&charge1+charge2<charge){
                charge=charge1+charge2;

            Get1(station[up][i].num,choose,'F',fb,'R',cr,Tool[station[up][i].num].forward[j]);
            }
        }
    }
}
}
else if(station[up][i].F_B=='B'){

```

```

        for(j=station[up][i].i+1; j<Tool[station[up][i].num].backward.size();
j++) {
            time2=charge2=Max;

            if(Change0(Tool[station[up][i].num].backward[j], down, choose, charge2, time2, fb, cr)) {
                if(station[up][i].C_R=='C') {
                    time1=(j-station[up][i].i)*CarRunTime;
                    charge1=RY(Tool[station[up][i].num].R_Y, j-station[up][i].i);
                    if(cr=='C') {
                        if(time1+time2+CarCarTime<time) {
                            time=time1+time2+CarCarTime;
                            charge=charge1+charge2;

                            Get1(station[up][i].num, choose, 'B', fb, 'C', cr, Tool[station[up][i].num].backward[j]);
                        }
                        else
                            if(time1+time2+CarCarTime==time&&charge1+charge2<charge) {
                                charge=charge1+charge2;

                                Get1(station[up][i].num, choose, 'B', fb, 'C', cr, Tool[station[up][i].num].backward[j]);
                            }
                        else if(cr=='R') {
                            if(time1+time2+CarRailTime<time) {
                                time=time1+time2+CarRailTime;
                                charge=charge1+charge2;

                                Get1(station[up][i].num, choose, 'B', fb, 'C', cr, Tool[station[up][i].num].backward[j]);
                            }
                            else
                                if(time1+time2+CarRailTime==time&&charge1+charge2<charge) {
                                    charge=charge1+charge2;

                                    Get1(station[up][i].num, choose, 'B', fb, 'C', cr, Tool[station[up][i].num].backward[j]);
                                }
                        }
                    }
                else if(station[up][i].C_R=='R') {
                    time1=(j-station[up][i].i)*RailRunTime;
                    if(cr=='C') {
                        charge1=3;
                        if(time1+time2+RailCarTime<time) {
                            time=time1+time2+RailCarTime;
                            charge=charge1+charge2;

                            Get1(station[up][i].num, choose, 'B', fb, 'R', cr, Tool[station[up][i].num].backward[j]);
                        }
                        else
                            if(time1+time2+RailCarTime==time&&charge1+charge2<charge) {
                                charge=charge1+charge2;

```



```

void Change2()
{
    int i, j, k, h, choose, charge=Max;
    char fb, cr;
    double time=Max;
    for(i=0; i<station[up].size(); i++) {
        if(station[up][i].F_B=='F') {
            for(j=station[up][i].i+1; j<Tool[station[up][i].num].forward.size(); j++) {
                for(k=0; k<station[down].size(); k++) {
                    if(station[down][k].F_B=='F') {
                        for(h=station[down][k].i-1; h>=0; h--) {
                            time2=charge2=Max;

                            if(Change0(Tool[station[up][i].num].forward[j], Tool[station[down][k].num].forward[
h], choose, charge2, time2, fb, cr)) {

                                if(station[up][i].C_R=='C' && station[down][k].C_R=='C') {
                                    time1=(j-station[up][i].i)*CarRunTime;
                                    time3=(station[down][k].i-h)*CarRunTime;
                                    charge1=RY(Tool[station[up][i].num].R_Y, j-
station[up][i].i);

                                    charge3=RY(Tool[station[down][k].num].R_Y, station[down][k].i-h);
                                    if(cr=='C') {
                                        if(time1+time2+time3+CarCarTime*2<time) {
                                            time=time1+time2+time3+CarCarTime*2;
                                            charge=charge1+charge2+charge3;

                                            Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'F', 'C', cr, 'C', Tool[sta
tion[up][i].num].forward[j], Tool[station[down][k].num].forward[h]);

                                                }
                                                else
                                if(time1+time2+time3+CarCarTime*2==time&&charge1+charge2+charge3<charge) {
                                    charge=charge1+charge2+charge3;

                                    Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'F', 'C', cr, 'C', Tool[sta
tion[up][i].num].forward[j], Tool[station[down][k].num].forward[h]);

                                        }
                                    }
                                    else if(cr=='R') {

                                        if(time1+time2+time3+CarRailTime+RailCarTime<time) {
                                            time=time1+time2+time3+CarCarTime*2;
                                            charge=charge1+charge2+charge3;

                                            Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'F', 'C', cr, 'C', Tool[sta
tion[up][i].num].forward[j], Tool[station[down][k].num].forward[h]);

                                                }
                                                else

```



```

if (time1+time2+time3+CarRailTime+RailCarTime==time&&charge1+charge2+charge3<charge) {
    charge=charge1+charge2+charge3;

    Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','C',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
    }
    }
    else
if(station[up][i].C_R=='C' &&station[down][k].C_R=='R') {
    time1=(j-station[up][i].i)*CarRunTime;
    time3=(station[down][k].i-h)*RailRunTime;
    charge1=RY(Tool[station[up][i].num].R_Y,j-
station[up][i].i);

    if(cr=='C')
        charge3=3;
    else
        charge3=0;
    if(cr=='C') {

        if(time1+time2+time3+CarCarTime+CarRailTime<time) {

            time=time1+time2+time3+CarCarTime+CarRailTime;
            charge=charge1+charge2+charge3;

            Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','C',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
            }
            else
if(time1+time2+time3+CarCarTime+CarRailTime==time&&charge1+charge2+charge3<charge) {
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','C',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
                }
                }
                else if(cr=='R') {

                    if(time1+time2+time3+CarRailTime+RailCarTime<time) {

                        time=time1+time2+time3+CarRailTime+RailCarTime;
                        charge=charge1+charge2+charge3;

                        Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','C',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
                        }
                        else
if(time1+time2+time3+CarRailTime+RailCarTime==time&&charge1+charge2+charge3<charge) {
                            charge=charge1+charge2+charge3;

                            Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','C',cr,'R',Tool[sta

```

```

tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
        }
    }
}
else
if(station[up][i].C_R=='R' &&station[down][k].C_R=='C') {
    time1=(j-station[up][i].i)*RailRunTime;
    time3=(station[down][k].i-h)*CarRunTime;
    if(cr=='C')
        charge1=3;
    else
        charge1=0;

    charge3=RY(Tool[station[down][k].num].R_Y,station[down][k].i-h);
    if(cr=='C') {

        if(time1+time2+time3+RailCarTime+CarCarTime<time) {

            time=time1+time2+time3+RailCarTime+CarCarTime;
            charge=charge1+charge2+charge3;

            Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','R',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
        }
        else
        if(time1+time2+time3+RailCarTime+CarCarTime==time&&charge1+charge2+charge3<charge) {
            charge=charge1+charge2+charge3;

            Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','R',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
        }
    }
    else if(cr=='R') {

        if(time1+time2+time3+RailRailTime+RailCarTime<time) {

            time=time1+time2+time3+RailRailTime+RailCarTime;
            charge=charge1+charge2+charge3;

            Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','R',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
        }
        else
        if(time1+time2+time3+RailRailTime+RailCarTime==time&&charge1+charge2+charge3<charge) {
            charge=charge1+charge2+charge3;

            Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','R',cr,'C',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
        }
    }
}
}
}

```

```

    }
    else{
        time1=(j-station[up][i].i)*RailRunTime;
        time3=(station[down][k].i-h)*RailRunTime;
        if(cr=='C')
            charge1=3;
        else
            charge1=0;
        if(cr=='C')
            charge3=3;
        else
            charge3=0;
        if(cr=='C'){

            if(time1+time2+time3+RailCarTime+CarRailTime<time){
                time=time1+time2+time3+RailCarTime+CarRailTime;
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','R',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
            }
            else
if(time1+time2+time3+RailCarTime+CarRailTime==time&&charge1+charge2+charge3<charge){
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','R',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
            }
            }
        else if(cr=='R'){

            if(time1+time2+time3+RailRailTime+RailRailTime<time){

                time=time1+time2+time3+RailRailTime+RailRailTime;
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','R',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
            }
            else
if(time1+time2+time3+RailRailTime+RailRailTime==time&&charge1+charge2+charge3<charge){
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'F','R',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].forward[h]);
            }
        }
    }
}
}
else if(station[down][k].F_B=='B'){

```

```

        for(h=station[down][k].i-1; h>=0; h--) {
            time2=charge2=Max;

            if(Change0(Tool[station[up][i].num].forward[j],Tool[station[down][k].num].backward
[h], choose, charge2, time2, fb, cr)) {

                if(station[up][i].C_R=='C' &&station[down][k].C_R=='C') {
                    time1=(j-station[up][i].i)*CarRunTime;
                    time3=(station[down][k].i-h)*CarRunTime;
                    charge1=RY(Tool[station[up][i].num].R_Y, j-
station[up][i].i);

                    charge3=RY(Tool[station[down][k].num].R_Y, station[down][k].i-h);
                    if(cr=='C') {
                        if(time1+time2+time3+CarCarTime*2<time) {
                            time=time1+time2+time3+CarCarTime*2;
                            charge=charge1+charge2+charge3;

                            Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'B', 'C', cr, 'C', Tool[sta
tion[up][i].num].forward[j], Tool[station[down][k].num].backward[h]);

                                }
                                else
if (time1+time2+time3+CarCarTime*2==time&&charge1+charge2+charge3<charge) {
                                    charge=charge1+charge2+charge3;

                                    Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'B', 'C', cr, 'C', Tool[sta
tion[up][i].num].forward[j], Tool[station[down][k].num].backward[h]);
                                        }
                                        }
                                        else if(cr=='R') {

                                            if(time1+time2+time3+CarRailTime+RailCarTime<time) {
                                                time=time1+time2+time3+CarCarTime*2;
                                                charge=charge1+charge2+charge3;

                                                Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'B', 'C', cr, 'C', Tool[sta
tion[up][i].num].forward[j], Tool[station[down][k].num].backward[h]);
                                                    }
                                                    else
if (time1+time2+time3+CarRailTime+RailCarTime==time&&charge1+charge2+charge3<charge) {
                                                        charge=charge1+charge2+charge3;

                                                        Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'B', 'C', cr, 'C', Tool[sta
tion[up][i].num].forward[j], Tool[station[down][k].num].backward[h]);
                                                            }
                                                            }
                                                            }
                                                            else
if (station[up][i].C_R=='C' &&station[down][k].C_R=='R') {
                        time1=(j-station[up][i].i)*CarRunTime;

```

```

        time3=(station[down][k].i-h)*RailRunTime;
        charge1=RY(Tool[station[up][i].num].R_Y,j-
station[up][i].i);
        if(cr=='C')
            charge3=3;
        else
            charge3=0;
        if(cr=='C'){

            if(time1+time2+time3+CarCarTime+CarRailTime<time){

                time=time1+time2+time3+CarCarTime+CarRailTime;
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','C',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
            }
            else
            if(time1+time2+time3+CarCarTime+CarRailTime==time&&charge1+charge2+charge3<charge){
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','C',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
            }
        }
        else if(cr=='R'){

            if(time1+time2+time3+CarRailTime+RailCarTime<time){

                time=time1+time2+time3+CarRailTime+RailCarTime;
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','C',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
            }
            else
            if(time1+time2+time3+CarRailTime+RailCarTime==time&&charge1+charge2+charge3<charge){
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','C',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
            }
        }
    }
    else
    if(station[up][i].C_R=='R' &&station[down][k].C_R=='C'){
        time1=(j-station[up][i].i)*RailRunTime;
        time3=(station[down][k].i-h)*CarRunTime;
        if(cr=='C')
            charge1=3;
        else

```

```

        charge1=0;

        charge3=RY(Tool[station[down][k].num].R_Y, station[down][k].i-h);
        if(cr=='C'){

            if(time1+time2+time3+RailCarTime+CarCarTime<time){

                time=time1+time2+time3+RailCarTime+CarCarTime;
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'B', 'R', cr, 'C', Tool[station[up][i].num].forward[j], Tool[station[down][k].num].backward[h]);
            }
            else
            if(time1+time2+time3+RailCarTime+CarCarTime==time&&charge1+charge2+charge3<charge){
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'B', 'R', cr, 'C', Tool[station[up][i].num].forward[j], Tool[station[down][k].num].backward[h]);
            }
            }
            else if(cr=='R'){

                if(time1+time2+time3+RailRailTime+RailCarTime<time){

                    time=time1+time2+time3+RailRailTime+RailCarTime;
                    charge=charge1+charge2+charge3;

                    Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'B', 'R', cr, 'C', Tool[station[up][i].num].forward[j], Tool[station[down][k].num].backward[h]);
                }
                else
                if(time1+time2+time3+RailRailTime+RailCarTime==time&&charge1+charge2+charge3<charge){
                    charge=charge1+charge2+charge3;

                    Get2(station[up][i].num, choose, station[down][k].num, 'F', fb, 'B', 'R', cr, 'C', Tool[station[up][i].num].forward[j], Tool[station[down][k].num].backward[h]);
                }
            }
        }
    }
    else{
        time1=(j-station[up][i].i)*RailRunTime;
        time3=(station[down][k].i-h)*RailRunTime;
        if(cr=='C')
            charge1=3;
        else
            charge1=0;
        if(cr=='C')
            charge3=3;
        else

```

```

        charge3=0;
        if(cr=='C'){

            if(time1+time2+time3+RailCarTime+CarRailTime<time){
                time=time1+time2+time3+RailCarTime+CarRailTime;
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','R',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
            }
            else
            if(time1+time2+time3+RailCarTime+CarRailTime==time&&charge1+charge2+charge3<charge){
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','R',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
            }
            else if(cr=='R'){

                if(time1+time2+time3+RailRailTime+RailRailTime<time){

                    time=time1+time2+time3+RailRailTime+RailRailTime;
                    charge=charge1+charge2+charge3;

                    Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','R',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
                }
                else
                if(time1+time2+time3+RailRailTime+RailRailTime==time&&charge1+charge2+charge3<charge){
                    charge=charge1+charge2+charge3;

                    Get2(station[up][i].num,choose,station[down][k].num,'F',fb,'B','R',cr,'R',Tool[sta
tion[up][i].num].forward[j],Tool[station[down][k].num].backward[h]);
                }
            }
        }
    }
}
else if(station[up][i].F_B=='B'){
    for(j=station[up][i].i+1; j<Tool[station[up][i].num].backward.size();
j++){
        for(k=0; k<station[down].size(); k++){
            if(station[down][k].F_B=='F'){
                for(h=station[down][k].i-1; h>=0; h--){
                    time2=charge2=Max;

                    if(Change0(Tool[station[up][i].num].backward[j],Tool[station[down][k].num].forward

```

```

[h], choose, charge2, time2, fb, cr)) {

    if(station[up][i].C_R=='C' &&station[down][k].C_R=='C') {
        time1=(j-station[up][i].i)*CarRunTime;
        time3=(station[down][k].i-h)*CarRunTime;
        charge1=RY(Tool[station[up][i].num].R_Y, j-
station[up][i].i);

        charge3=RY(Tool[station[down][k].num].R_Y, station[down][k].i-h);
        if(cr=='C') {
            if(time1+time2+time3+CarCarTime*2<time) {
                time=time1+time2+time3+CarCarTime*2;
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'F', 'C', cr, 'C', Tool[sta
tion[up][i].num].backward[j], Tool[station[down][k].num].forward[h]);

            }
            else
if(time1+time2+time3+CarCarTime*2==time&&charge1+charge2+charge3<charge) {
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'F', 'C', cr, 'C', Tool[sta
tion[up][i].num].backward[j], Tool[station[down][k].num].forward[h]);

            }
        }
        else if(cr=='R') {

            if(time1+time2+time3+CarRailTime+RailCarTime<time) {
                time=time1+time2+time3+CarCarTime*2;
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'F', 'C', cr, 'C', Tool[sta
tion[up][i].num].backward[j], Tool[station[down][k].num].forward[h]);

            }
            else
if(time1+time2+time3+CarRailTime+RailCarTime==time&&charge1+charge2+charge3<charge) {
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'F', 'C', cr, 'C', Tool[sta
tion[up][i].num].backward[j], Tool[station[down][k].num].forward[h]);

            }
        }
        else
if(station[up][i].C_R=='C' &&station[down][k].C_R=='R') {
            time1=(j-station[up][i].i)*CarRunTime;
            time3=(station[down][k].i-h)*RailRunTime;
            charge1=RY(Tool[station[up][i].num].R_Y, j-
station[up][i].i);

            if(cr=='C')

```



```

        charge3=3;
    else
        charge3=0;
    if(cr=='C') {

        if(time1+time2+time3+CarCarTime+CarRailTime<time) {

            time=time1+time2+time3+CarCarTime+CarRailTime;
            charge=charge1+charge2+charge3;

            Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','C',cr,'R',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);
        }
        else
        if(time1+time2+time3+CarCarTime+CarRailTime==time&&charge1+charge2+charge3<charge) {
            charge=charge1+charge2+charge3;

            Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','C',cr,'R',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);
        }
    }
    else if(cr=='R') {

        if(time1+time2+time3+CarRailTime+RailCarTime<time) {

            time=time1+time2+time3+CarRailTime+RailCarTime;
            charge=charge1+charge2+charge3;

            Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','C',cr,'R',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);
        }
        else
        if(time1+time2+time3+CarRailTime+RailCarTime==time&&charge1+charge2+charge3<charge) {
            charge=charge1+charge2+charge3;

            Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','C',cr,'R',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);
        }
    }
}
else
if(station[up][i].C_R=='R' && station[down][k].C_R=='C') {
    time1=(j-station[up][i].i)*RailRunTime;
    time3=(station[down][k].i-h)*CarRunTime;
    if(cr=='C')
        charge1=3;
    else
        charge1=0;

    charge3=RY(Tool[station[down][k].num].R_Y,station[down][k].i-h);
    if(cr=='C') {

```

```

if(time1+time2+time3+RailCarTime+CarCarTime<time) {

time=time1+time2+time3+RailCarTime+CarCarTime;
charge=charge1+charge2+charge3;

Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','R',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);
}
else
if(time1+time2+time3+RailCarTime+CarCarTime==time&&charge1+charge2+charge3<charge) {
charge=charge1+charge2+charge3;

Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','R',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);
}
}
else if(cr=='R') {

if(time1+time2+time3+RailRailTime+RailCarTime<time) {

time=time1+time2+time3+RailRailTime+RailCarTime;
charge=charge1+charge2+charge3;

Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','R',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);
}
else
if(time1+time2+time3+RailRailTime+RailCarTime==time&&charge1+charge2+charge3<charge) {
charge=charge1+charge2+charge3;

Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'F','R',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].forward[h]);
}
}
}
else{
time1=(j-station[up][i].i)*RailRunTime;
time3=(station[down][k].i-h)*RailRunTime;
if(cr=='C')
charge1=3;
else
charge1=0;
if(cr=='C')
charge3=3;
else
charge3=0;
if(cr=='C') {

if(time1+time2+time3+RailCarTime+CarRailTime<time) {

```



```

        time=time1+time2+time3+CarCarTime*2;
        charge=charge1+charge2+charge3;

        Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','C',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);

    }
    else
if (time1+time2+time3+CarCarTime*2==time&&charge1+charge2+charge3<charge) {
        charge=charge1+charge2+charge3;

        Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','C',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
    }
}
else if (cr=='R') {

    if (time1+time2+time3+CarRailTime+RailCarTime<time) {
        time=time1+time2+time3+CarCarTime*2;
        charge=charge1+charge2+charge3;

        Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','C',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
    }
    else
if (time1+time2+time3+CarRailTime+RailCarTime==time&&charge1+charge2+charge3<charge) {
        charge=charge1+charge2+charge3;

        Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','C',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
    }
}
}
else
if (station[up][i].C_R=='C' && station[down][k].C_R=='R') {
        time1=(j-station[up][i].i)*CarRunTime;
        time3=(station[down][k].i-h)*RailRunTime;
        charge1=RY(Tool[station[up][i].num].R_Y,j-
station[up][i].i);

        if (cr=='C')
            charge3=3;
        else
            charge3=0;
        if (cr=='C') {

            if (time1+time2+time3+CarCarTime+CarRailTime<time) {

                time=time1+time2+time3+CarCarTime+CarRailTime;
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','C',cr,'R',Tool[sta

```

```

tion[up][i].num].backward[j], Tool[station[down][k].num].backward[h]);
    }
    else
if (time1+time2+time3+CarCarTime+CarRailTime==time&&charge1+charge2+charge3<charge) {
    charge=charge1+charge2+charge3;

    Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'B', 'C', cr, 'R', Tool[sta
tion[up][i].num].backward[j], Tool[station[down][k].num].backward[h]);
    }
}
else if (cr=='R') {

if (time1+time2+time3+CarRailTime+RailCarTime<time) {

time=time1+time2+time3+CarRailTime+RailCarTime;
    charge=charge1+charge2+charge3;

    Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'B', 'C', cr, 'R', Tool[sta
tion[up][i].num].backward[j], Tool[station[down][k].num].backward[h]);
    }
    else
if (time1+time2+time3+CarRailTime+RailCarTime==time&&charge1+charge2+charge3<charge) {
    charge=charge1+charge2+charge3;

    Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'B', 'C', cr, 'R', Tool[sta
tion[up][i].num].backward[j], Tool[station[down][k].num].backward[h]);
    }
}
}
else
if (station[up][i].C_R=='R' && station[down][k].C_R=='C') {
    time1=(j-station[up][i].i)*RailRunTime;
    time3=(station[down][k].i-h)*CarRunTime;
    if (cr=='C')
        charge1=3;
    else
        charge1=0;

    charge3=RY(Tool[station[down][k].num].R_Y, station[down][k].i-h);
    if (cr=='C') {

if (time1+time2+time3+RailCarTime+CarCarTime<time) {

time=time1+time2+time3+RailCarTime+CarCarTime;
    charge=charge1+charge2+charge3;

    Get2(station[up][i].num, choose, station[down][k].num, 'B', fb, 'B', 'R', cr, 'C', Tool[sta
tion[up][i].num].backward[j], Tool[station[down][k].num].backward[h]);
    }
    else
if (time1+time2+time3+RailCarTime+CarCarTime==time&&charge1+charge2+charge3<charge) {

```

```

charge=charge1+charge2+charge3;

Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','R',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
    }
}
else if(cr=='R'){

if(time1+time2+time3+RailRailTime+RailCarTime<time){

time=time1+time2+time3+RailRailTime+RailCarTime;
charge=charge1+charge2+charge3;

Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','R',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
    }
else
if(time1+time2+time3+RailRailTime+RailCarTime==time&&charge1+charge2+charge3<charge){
charge=charge1+charge2+charge3;

Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','R',cr,'C',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
    }
}
}
else{
time1=(j-station[up][i].i)*RailRunTime;
time3=(station[down][k].i-h)*RailRunTime;
if(cr=='C')
charge1=3;
else
charge1=0;
if(cr=='C')
charge3=3;
else
charge3=0;
if(cr=='C'){

if(time1+time2+time3+RailCarTime+CarRailTime<time){
time=time1+time2+time3+RailCarTime+CarRailTime;
charge=charge1+charge2+charge3;

Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','R',cr,'R',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
    }
else
if(time1+time2+time3+RailCarTime+CarRailTime==time&&charge1+charge2+charge3<charge){
charge=charge1+charge2+charge3;

Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','R',cr,'R',Tool[sta

```

```

tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
        }
    }
    else if(cr=='R'){

        if(time1+time2+time3+RailRailTime+RailRailTime<time){

            time=time1+time2+time3+RailRailTime+RailRailTime;

            Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','R',cr,'R',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
        }
        else
            if(time1+time2+time3+RailRailTime+RailRailTime==time&&charge1+charge2+charge3<charge){
                charge=charge1+charge2+charge3;

                Get2(station[up][i].num,choose,station[down][k].num,'B',fb,'B','R',cr,'R',Tool[sta
tion[up][i].num].backward[j],Tool[station[down][k].num].backward[h]);
            }
        }
    }
}

if(time!=Max){
    if(cr1=='C')
        outfile<<"从S"<<up<<"到S"<<down<<"，你可以先乘坐L"<<tool1;
    else
        outfile<<"从S"<<up<<"到S"<<down<<"，你可以先乘坐T"<<tool1-CarNum;
    if(at1<=StationNum)
        outfile<<"在S"<<at1<<"换乘";
    else
        outfile<<"在D"<<at1-StationNum<<"换乘";
    if(cr2=='C')
        outfile<<"L"<<tool2;
    else
        outfile<<'T'<<tool2-CarNum;
    if(at2<=StationNum)
        outfile<<"再在S"<<at2<<"换乘";
    else
        outfile<<"再在D"<<at2-StationNum<<"换乘";
    if(cr3=='C')
        outfile<<"L"<<tool3<<endl;
    else
        outfile<<'T'<<tool3-CarNum<<endl;
    outfile<<"所花时间为:"<<time<<ends<<"所需费用为:"<<charge<<endl;
    if(cr1=='C')
        outfile<<"L"<<tool1<<"行车路线为:";
}

```

```

        else
            outfile<<"D"<<tool1-CarNum<<"行车路线为:";
        Output0(up, at1, tool1, fb1, cr1);
        if (cr2=='C')
            outfile<<endl<<"L"<<tool2<<"行车路线为:";
        else
            outfile<<endl<<"D"<<tool2-CarNum<<"行车路线为:";
        Output0(at1, at2, tool2, fb2, cr2);
        if (cr3=='C')
            outfile<<endl<<"L"<<tool3<<"行车路线为:";
        else
            outfile<<endl<<"D"<<tool3-CarNum<<"行车路线为:";
        Output0(at2, down, tool3, fb3, cr3);
        outfile<<endl<<endl;
    }else
        outfile<<"从S"<<up<<"到S"<<down<<"，你不能通过换乘二次来达到目的地"<<endl<<endl;
}

int main()
{
    Data_Input();
    Data1_Input();
    Go_Input();
    int choose_tool;
    int charge=Max;
    char fb, cr;
    double time=Max;
    if (Change0(up, down, choose_tool, charge, time, fb, cr)) {
        if (cr=='C')
            outfile<<"从S"<<up<<"到S"<<down<<"，你可以乘坐直达车L"<<choose_tool<<endl;
        else
            outfile<<"从S"<<up<<"到S"<<down<<"，你可以乘坐地铁T"<<choose_tool-
CarNum<<endl;
        outfile<<"所花时间为:"<<time<<ends<<"所需费用为:"<<charge<<endl;
        outfile<<"行车路线为: ";
        Output0(up, down, choose_tool, fb, cr);
        outfile<<endl<<endl;
    }
    else{
        outfile<<"从S"<<up<<"到S"<<down<<"无直达车和地铁"<<endl<<endl;
    }
    Change1();
    Change2();
    return 0;
}

```