

2016 年全国青少年信息学奥林匹克竞赛 江苏省省队第一轮选拔赛

JSTSC 2016 第二试

竞赛时间：2016 年 4 月 17 日 8:00–13:00

题目名称	飞机调度	无界单词	轻重路径
源文件名	flight.c/cpp/pas	word.c/cpp/pas	heavy.c/cpp/pas
可执行文件名	flight.exe	word.exe	heavy.exe
输入文件名	flight.in	word.in	heavy.in
输出文件名	flight.out	word.out	heavy.out
每个测试点时限	1 秒	1 秒	3 秒
内存限制	512M	512M	512M
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	否	否	否
题目类型	传统型	传统型	传统型

提交源程序须加后缀

对于 Pascal 语言	flight.pas	word.pas	heavy.pas
对于 C 语言	flight.c	word.c	heavy.c
对于 C++ 语言	flight.cpp	word.cpp	heavy.cpp

注意：最终测试时，所有语言编译均不打开任何优化开关

飞机调度

【故事背景】

作为一个旅行达人以及航空公司的金卡会员，JYY 每一年的飞行里程可以绕赤道几周了。JYY 发现，航空公司为了提高飞机的使用率，并不是简单的一条航线使用一架飞机来回飞，而是会让同一架飞机连续不停的飞不同的航线，甚至有的时候为了能够完成飞机的调度，航空公司还会增开一些临时航线——在飞机转场的同时顺路捎一些乘客。JYY 研究了一下 JSOI 著名航空公司 JS Airways 的常规直飞航线，他想计算一下，在最佳调度方案下，JS Airway 最少需要多少架飞机才能成功执飞这所有的航线。

【问题描述】

JSOI 王国里有 N 个机场，编号为 1 到 N 。从 i 号机场到 j 号机场需要飞行 $T_{i,j}$ 的时间。由于风向，地理位置和航空管制的因素， $T_{i,j}$ 和 $T_{j,i}$ 并不一定相同。

此外，由于飞机降落之后需要例行维修和加油。当一架飞机降落 k 号机场时，需要花费 P_k 的维护时间才能再次起飞。

JS Airways 一共运营 M 条航线，其中第 i 条直飞航线需要在 D_i 时刻从 X_i 机场起飞，不经停，飞往 Y_i 机场。

为了简化问题，我们假设 JS Airway 可以在 0 时刻在任意机场布置任意多架加油维护完毕的飞机；为了减少飞机的使用数，我们允许 JS Airways 增开任意多条临时航线以满足飞机的调度需求。

JYY 想知道，理论上 JS Airways 最少需要多少架飞机才能完成所有这 M 个航班。

【输入格式】

从文件 *flight.in* 中读入数据。

输入一行包含两个正整数 N 和 M 。

接下来一行包含 N 个正整数表示每一个机场的飞机维护时间。

接下来 N 行，每行 N 个非负整数，其中第 i 行第 j 个非负整数为 $T_{i,j}$ ，表示从 i 号机场飞往 j 号机场所需要花费的时间。数据保证 $T_{i,i} = 0$ 。

接下来 M 行，每行 3 个正整数，其中第 i 行为 X_i, Y_i, D_i ，表示第 i 条航线的起飞机场，降落机场，以及起飞时间。数据保证 $X_i \neq Y_i$ 。

【输出格式】

输出到文件 *flight.out* 中。

输出文件包含一行一个正整数，表示 JS Airways 理论上最少需要的飞机数。

【输入样例 1】

```
3 3
100 1 1
0 1 1
1 0 5
2 1 0
1 2 1
2 1 1
3 1 9
```

【输出样例 1】

```
2
```

【输入样例 2】

```
3 3
100 1 1
0 1 1
1 0 5
2 1 0
1 2 1
2 1 1
3 1 8
```

【输出样例 2】

```
3
```

【样例说明】

在第一个样例中，JS Airways 可以在 0 时刻在 2 号机场安排一架飞机并执飞第 2 条航线（2→1）。此外还需要在 0 时刻在 1 号机场安排一架飞机，这架飞机首先执飞第 1 条航线（1→2），然后通过临时新增一条航线从 2 号机场起飞飞往 3 号机场，降落 3 号机场之后执飞第 3 条航线（3→1）。

在第二个样例中，执行完第 1 条航线的飞机无法赶上第 3 条航线的起飞时间，因此 JS Airways 必须使用 3 架不同的飞机才能完成所有的航班。

【数据规模】

对于 30% 的数据满足 $N, M \leq 10$;

对于 60% 的数据满足 $N, M \leq 100$;

对于 100% 的数据满足 $1 \leq N, M \leq 500, 0 \leq P_i, T_{i,j} \leq 10^6, 1 \leq D_i \leq 10^6$ 。

无界单词

【故事背景】

JYY 在坐飞机的时候总是喜欢随便写点文字以打发时间。

【问题描述】

对于一个单词 S ，如果存在一个长度 l ，满足 $0 < l < \text{length}(S)$ ，并且使得 S 长度为 l 的前缀与 S 长度为 l 的后缀相同，JYY 则称 S 是有界的。比如“aabaa”和“ababab”就都是有界的字符串。如果一个单词不存在这样的 l ，则 JYY 称之为无界单词。

现在考虑所有仅由字母 a 和 b 组成的长度为 N 的字符串，JYY 想知道

- (1) 一共有多少个无界单词？
- (2) 这些无界单词中，按字典序排列第 K 小的单词是哪一个？

【输入格式】

从文件 `word.in` 中读入数据。一个文件中包含多组测试数据。

第一行包含一个正整数 T ，表示测试数据的组数；

接下来 T 行，每行包含两个正整数 N 和 K ，表示一组测试数据。

【输出格式】

输出到文件 `word.out` 中。

对于每一个测试数据，输出两行。

其中第一行包含一个整数，表示长度为 N 的无界单词的数量；

其中第二行包含一个长度为 N 的字符串，表示第 K 小的无界单词。

【样例输入】

```
5
5 1
5 2
5 3
5 4
5 5
```

【样例输出】

```
12
```

aaaab
12
aaabb
12
aabab
12
aabbb
12
ababb

【数据规模与约定】

对于 20% 的数据满足 $N \leq 20$;

对于 100% 的数据满足 $1 \leq T \leq 5, 1 \leq N \leq 64$ 。

输入数据保证，对于任意测试数据，总存在第 K 小的无界单词。

轻重路径

【故事背景】

JYY 最近学习了一种处理树形结构的高级技巧,叫“轻重路径剖分”。这种技术会将树中的边划分成轻边和重边。相连的重边会形成一些树上相离的路径。“轻重路径剖分”可以使得从树上任意一点走到根,都至多只会经过 $O(\log N)$ 条不同的重路径。

【问题描述】

如果你不了解轻重路径剖分, JYY 在这里简单介绍一下:

对于一棵有根树中的任意一个点 u , 我们用 $\text{size}(u)$ 表示其为根的子树中的点的数量。对于 u 的所有孩子中, 我们选出 $\text{size}(\cdot)$ 值最大的孩子 v , 并将边 $\langle u, v \rangle$ 设置成重边, u 和其他孩子之间的边我们均设置为轻边。

为了简化问题, 这里 JYY 仅考虑一颗 N 个点的有根二叉树。这 N 个点由 1 到 N 编号。并且如果 u 存在两个 $\text{size}(\cdot)$ 值一样的孩子, 则我们默认 u 和其左孩子的连边为重边。

现在 JYY 希望执行额外 Q 次删点操作, 每次 JYY 会随机删掉一个当前二叉树的叶子节点, 而你则需要动态的维护这棵树的轻重路径剖分。

为了方便输出, 你只需要在每次操作后输出所有重边指向的点的权值和即可。

如果删除一个点之后, 存在一个点 u 拥有两个 $\text{size}(\cdot)$ 值一样的孩子, 则我们保持 u 在该操作执行之前的重边划分。

【输入格式】

从文件 *heavy.in* 中读入数据。

输入文件的第一行包含一个整数 N 。

接下来 N 行, 第 i 行包含两个整数 L_i, R_i , 表示编号为 i 的点的左孩子编号和右孩子编号; $L_i = 0$ 表示点 i 没有左孩子, $R_i = 0$ 表示点 i 没有右孩子。

第 $N+2$ 行包含一个整数 Q , 表示 JYY 进行的删点操作。

第 $N+3$ 行包含 Q 个空格分开的正整数, 表示 JYY 删去的叶子的编号。

输入数据保证每次删除操作均删除了一个叶子。

【输出格式】

输出到文件 *heavy.out* 中。

输出 $Q+1$ 行, 每行包含一个整数, 表示在轻重路径剖分中所有重边指向的点的编号的和。其中第一行对应初始的路径剖分, 之后的 Q 行对应进行了相应删点操作之后路径划分。

【样例输入】

```
8
2 3
4 5
0 0
6 7
0 8
0 0
0 0
0 0
7
6 7 8 5 4 2 3
```

【样例输出】

```
20
21
15
7
6
2
3
0
```

【数据规模】

对于 30%的数据满足 $N \leq 1,000$;
对于 50%的数据满足 $N \leq 50,000$;
对于 100%的数据满足 $N \leq 200,000$ 。