

作业 2

在线测试：<http://thudsa.3322.org/oj>

第 1 题	股票 (Stock)	20 分	1
第 2 题	01 博弈 (Zerone)	20 分	3
第 3 题	目录树 (Dirtree)	20 分	4
第 4 题	月亮和五毛钱 (Mooney)	20 分	5
第 5 题	贡多拉 (Gondola)	20 分	7

【说明】

- 1 除非特别声明，输入、输出各行内的（整数、浮点数、字符、字符串等）数据项之间，均以
一个空格相互分隔。

第1题 股票 (Stock)

20 分

【题目描述】

“天下大势，分久必合，合久必分”，股市亦是如此：新股不断发行，旧股相继退出。从发行到退出之间的时段，称作股票的活跃期。

为简化起见，这里不妨假定活跃期均以交易日为基本单位，并以股市开市之（第零）日作为统一起点。比如，若股票 A “在第 123 天发行并于第 130 天退出”，则其活跃期为第 123 至 129 天共计 7 天，记作 $[123, 130)$ 。另外，不妨假定各股票在其活动期内价格不变——否则，可等效于原股票退出，同时发行一支定价更新的股票。

Q 王国的股市虽不能脱俗，却亦有其与众不同之处。在该国的股市中，禁止不同股票的活跃期相互包含。比如，无论是 $[100, 150)$ 和 $[120, 130)$ 、 $[120, 130)$ 和 $[120, 150)$ 、 $[100, 130)$ 和 $[120, 130)$ ，甚至是 $[120, 130)$ 和 $[120, 130)$ ，都不会同时作为股票的活跃期。也就是说，每天发行和退市的股票累计不超过一支。

Maximum (卖个死萌) 先生是 Q 王国公认的股神，他的策略既稳妥也简单。所谓稳妥是指，任何时候的投资总额都保持固定，如此可保证风险 (risk) 有限。所谓简单是指，每天都将所有的投资集中用于购入当日价格最低的一支股票（为此可能需要先卖出原持有股票，且交易所需时间忽略不计），如此他的持股量每天都能保持最大——在 Q 王国，这可是评估股市收益 (return) 的首要指标。

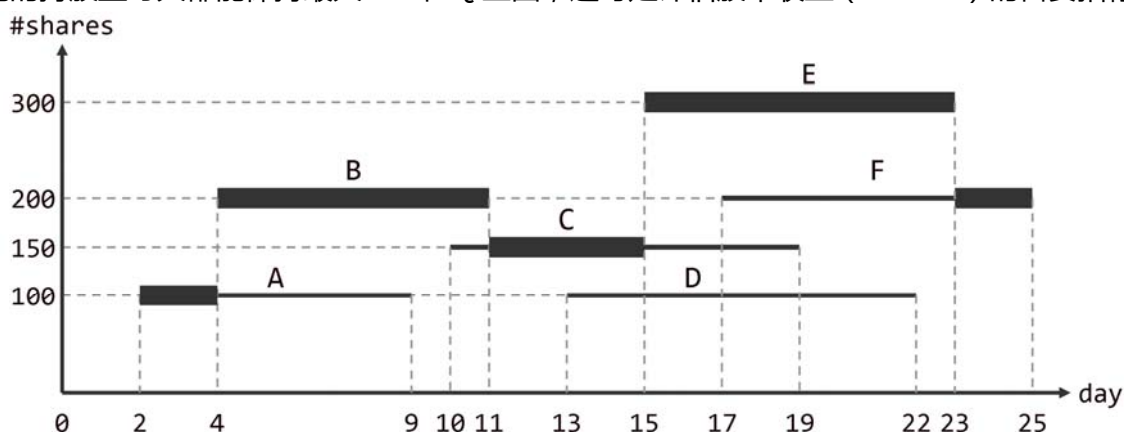


图1. 实例：股市共持续25天，先后发行6支股票，Maximum先生日均持有200股

不过，尽管 Maximum 先生十分希望精确地知道自己在一段时期内的日均持股量，但对于此类计算他却毫无自信。于是终于有一天，他揣着长长的一叠股市历史记录，敲响了你的家门。

【输入】

第 1 行为 1 个正整数，即曾经发行过的股票总数 N 。

接下来的 $2N$ 行，按照日期升序排列地描述股票的历史，以下两种格式的事件各半：

D X //在上一事件之后的第D天有一支股票发行，且按Maximum先生的投资额可购入X股

D //在上一事件之后的第D天有一支股票退市

其中，除首个事件的 D 值可能为 0，其余 D 值及 X 值均为正整数。

输入的时间序列合法，即保证活跃股票数始终非负。

【输出】

自股市开市至最后一支股票退出期间，Maximum 先生的日均持股量（四舍五入精确至百分位）。

【输入样例】

```
6
2 100
2 200
5
1 150
1
2 100
2 300
2 200
2
3
1
2
```

【输出样例】

```
200.00
```

【限制】

$1 \leq N < 10^7$

$1 \leq X < 10^3$

股市的营业期不超过 $2^{31} - 1$ 天

【提示】

在队列结构的基础上，扩充操作接口

第2题 01 博弈 (Zerone)**20 分****【题目描述】**

Adam 和 Eve 被赶出伊甸园后，盖起了四面高墙。高墙上写着 n 行 01 串。两人无事可做，于是开始就这些 01 串做如下博弈：

1. Adam进行第1手，此后双方轮流操作。
2. 第 i 手的操作者，可以且必须在0和1之间选择，并相应地抹掉某些串。具体地，若选择0 (1)，则抹掉第 i 位为0 (1) 的所有串。长度短于 i 的串，也须抹掉。
3. 一方操作之后若将所有串都抹掉了，则判该方失败。

不难看出，若两人皆明智，则胜负必然**确定**。不幸的是，他俩虽明智却更懒惰，不愿按部就班地计算却又迫切地想知道：对于每个 i ，倘若对前 i 个串进行博弈，谁将获胜。请你本着帮舍友找对象的精神，写个程序帮帮他俩。

【输入】

第 1 行含一个正整数，即初始写在墙上的 01 串总数 n 。随后的 n 行，依次给出第 1~ n 个 01 串。

【输出】

不超过 n 行，各由三部分组成：首先是 "Adam" 或 "Eve"，代表必胜方；接着是正整数 $start$ 和 end ，表示必胜的结果从前 $start$ 行持续到前 end 行。

各行按 $start$ 值递增输出，且相邻行的必胜方互异。

【输入样例】

```
10
10101
1011
1110
000
01
110
10110
1001
11
0010
```

【输出样例】

```
Adam 1 1
Eve 2 3
Adam 4 4
Eve 5 6
Adam 7 10
```

【限制】

$1 \leq n \leq 10^6$

$1 \leq \text{01 串长度} \leq 64$

输入数据不超过 35MB

【提示】

二叉编码树

第3题 目录树 (Dirtree)**20 分**

任给某个文件系统中的一组路径，试整理并输出覆盖这些路径的最小文件系统目录结构。

【输入】

第 1 行为一个整数，即所给路径的总数 n 。

以下 n 行各对应于一条路径。目录名由英文字母、数字、"."、"_"、"-"组成，各层目录名以"/"分隔，路径长度均不超过 255 个字符。

【输出】

以先序遍历序列形式给出的目录结构。

同层目录按字典序排列。每深入一层，目录名前须增加 2 个空格的缩进。

【输入样例】

```
6
usr/bin
usr/local/bin
bin
usr/share
usr/local/share
zip
```

【输出样例】

```
bin
usr
  bin
  local
    bin
    share
share
zip
```

【限制】

$1 \leq n \leq 10^5$

输入数据不超过 30MB

【提示】

无

第4题 月亮和五毛钱 (Mooney)**20 分****【题目描述】**

这是一个宁静的夜晚，月光洒在地上的五毛钱上。

古人有云：“举头望明月，不见地上钱”。这句话的意思是，倘若只顾着遥望空中的月亮，便会看不见地上反射月光的一枚枚五毛钱。既然五毛钱和月亮都是反射了它光而发光的圆形物体，那它们又有什么区别呢？

高更 (GG) 的目光如飞蛾，总是扑向发光的物体。夜里行路，在每一个交叉路口借光看方向时，他总会寻找光源：要么抬头望月，要么低头找钱。不过他觉得仰着脖子太累，故希望能在通往目的地的沿途，**尽量少**看月亮。

毛姆 (MM) 的追求则有所不同。他希望能在通往目的地的沿途，捡起**尽量多**的钱——注意，钱是掉在地上的，而不是从地里源源不断长出来的，因此不能重复捡起。

既然是数据结构的作业，故城市不妨化身作图——一张有向图。其中，节点对应于交叉路口，边对应于路口之间的道路——每个路口都可连接任意多条道路，而道路则都是单行线。显然，月亮如影，总在头顶，处处可以望见；而五毛钱，却只在某些交叉路口出没。

现在给出这个城市的地图，请你通过程序计算出，按他们各自的追求，高更**最少**要看多少次月亮，毛姆**最多**能捡多少五毛钱。

【输入】

第 1 行为两个正整数，分别为城市有向图的点数 n 和边数 m 。

第 2 行为一个长度为 n 的字符串，依次给出各点（即编号 $[0, n)$ 的各路口）的路况：'M' 表示此处只有月亮，'m' 表示还有五毛钱。

接下来的 m 行分别描述各边。各含两个 $[0, n)$ 之间的整数 a 和 b ，表示从点 a 到点 b 有一条边。

高更和毛姆的起点都是点 0 ，目的地都是点 $n-1$ 。保证从起点到终点有通路相联。

【输出】

第 1 行一个正整数，表示高更**最少**要看多少次月亮。

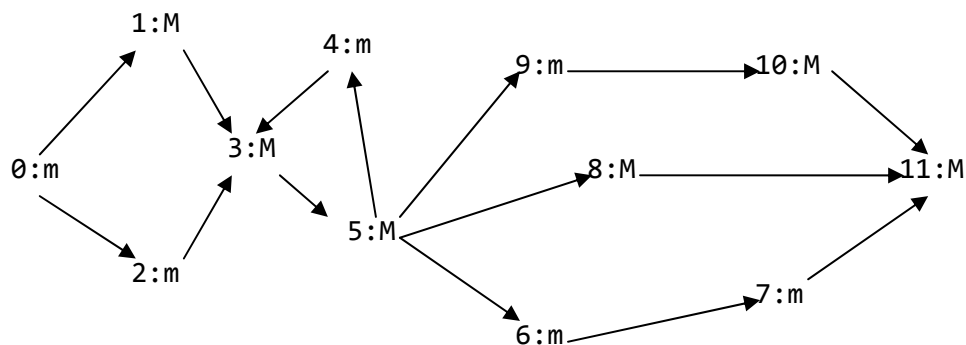
第 2 行一个正整数，表示毛姆**最多**能捡多少五毛钱。

【输入样例】

```
12 15
mMmMmMmmMmMM
0 1
0 2
1 3
2 3
4 3
3 5
5 4
5 6
5 8
6 7
8 11
7 11
5 9
9 10
10 11
```

【输出样例】

```
3
5
```



【限制】

$1 \leq n \leq 500,000$
 $1 \leq m \leq 1,200,000$

【提示】

BFS，强连通分量，拓扑序动态规划

第5题 贡多拉 (Gondola)**20 分****【题目描述】**

素有“水城”之称的威尼斯由数不胜数的水道、小岛和桥梁组成,它也是世界上唯一不通汽车的城市,船是唯一的交通工具。

贡多拉是威尼斯特有的和最具代表性的传统划船,船身全漆黑色,由一船夫站在船尾划动。随着科技的进步,水上巴士已经替代了贡多拉成为人们日常出行的主要交通工具,而贡多拉则作为旅游业用途而保留下来了。乘坐贡多拉游览威尼斯的美景是一种惬意的事情,当然也是一种奢侈的事情。



为更好地吸引游客,贡多拉协会开始提供网上预约服务,并根据游客的需要定制游览线路。线路设计最初由人工完成,但是随着这项服务受到越来越多游客的青睐,这一工作的压力越来越大。于是,协会会长只好求助于身为计算机专家的你。

威尼斯的水路网由水道和连接不同水道的交叉点组成。每条游览路线,就是若干条水道串接而成的序列,从一条水道进入下一条水道必然会经过某个交叉点。为简单起见,假设客人们选择的路线起点和终点都在交叉点上。

根据此前导游的经验和游客的反馈,贡多拉协会已对威尼斯的各处景点做过评估。不过由于景点自身特点和游客偏好的不同,评估结果逐渐构成了一个多维度的评分系统。比如,叹息桥的评分是(历史=20, 浪漫=30),圣马可大教堂的评分是(历史=30, 宗教=30, 艺术=20)。而按照这个评分系统,游客们的需求也抽象成了多个维度的分值。比如,若某游客要求历史总分 40,则让他参观叹息桥和圣马可大教堂即可——这些景点在历史方面的总分是 $20+30 = 50 \geq 40$ 。

威尼斯遍地是风景,威尼斯的每一条水道,每一个交叉点都有其独特的风景,协会也对这些风景做过整理,形成了一份每条水道和每个交叉点的评分表。线路的设计就是根据该评分表进行计算,每一条路线在每一维度的得分,为该线路所经水道和交叉点该维度的分数之和。特别地,不会有游客喜欢耽误工夫故地重游,所以再次经过已游历过的水道和交叉点,都不会获得任何分数(会长:不扣你分就不错了)。另外,做起点和终点的交叉点处的得分,也不计入游览路线的得分。

对游客来说,乘坐贡多拉游览是按时计费的且价格不菲;而对协会来说,他们希望接待尽可能多的游客。因此,协会会长希望你的程序所设计出的游览路线,能够尽可能短。

【输入】

第 1 行含三个正整数,分别为城市的交叉点总数 n 、水道总数 m 以及评分系统的维数 k 。

接下来的 n 行描述各交叉点。各含 k 个非负整数,即所对应交叉点的 k 项评分。

接下来的 m 行描述各条水道。各含 $k+3$ 个非负整数:前三个分别为该水道的长度 l 、水道两端所连接的交叉点编号 s 和 e ($0 \leq \{s, e\} < n$);后 k 个为该水道的 k 项评分。

最后一行是游客的路线设计需求,包含 $k+2$ 个非负整数:前两个是起点和终点的交叉点编号,后 k 个整数是各维度的游览得分需求。

输入数据保证至少有可行方案。

【输出】

一行，即针对该游客所提需求而设计出的游览路线，格式如下：

交叉点 水道 交叉点

其中，交叉点为 "c*" 形式 ("*" 为交叉点的编号，比如 0 号交叉点就是 "c0"); 水道为 "R*" 形式 ("*" 为水道的编号，比如 12 号水道就是 "R12")。

【输入样例】

```
4 6 3
0 10 0
10 5 5
0 0 30
5 5 0
10 0 1 30 0 0
8 1 2 0 0 5
7 3 2 10 10 0
8 3 0 0 10 0
12 0 2 5 5 5
7 0 2 0 0 0
0 2 55 0 0
```

【输出样例】

```
C0 R0 C1 R1 C2 R2 C3 R3 C0 R5 C2
```

【限制】

$$2 \leq n \leq 100$$

$$1 \leq m \leq 10,000$$

$$1 \leq k \leq 20$$

每个测试点的时限为 $\lceil (n*m)^{1/6} \rceil$ 秒。

【评分】

鉴于此问题的难度，将在方案可行的前提下，根据你所设计方案的优化程度（反比于游览路线的长度），按全班排名评定各测例的得分。因此，你的最终得分可能与平时 OJ 的评测报告不完全一致。

请尽可能优化你的算法，以争取获得更好的成绩。

【提示】

非最优化算法，搜索剪枝