

# CCF NOI 2018 省队选拔赛

## 第二试

时间：2018 年 4 月 7 日 08:00 ~ 12:30

题目名称	劈配	林克卡特树	制胡窜
题目类型	传统型	传统型	传统型
目录	mentor	lct	cutting
可执行文件名	mentor	lct	cutting
输入文件名	mentor.in	lct.in	cutting.in
输出文件名	mentor.out	lct.out	cutting.out
每个测试点时限	1.0 秒	12.0 秒	5.0 秒
内存限制	512 MB	1 GB	512 MB
测试点数目	10	20	20
每个测试点分值	10	5	5

提交源程序文件名

对于 C++ 语言	mentor.cpp	lct.cpp	cutting.cpp
对于 C 语言	mentor.c	lct.c	cutting.c
对于 Pascal 语言	mentor.pas	lct.pas	cutting.pas

编译选项

对于 C++ 语言	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11
对于 C 语言	-O2 -std=c11	-O2 -std=c11	-O2 -std=c11
对于 Pascal 语言	-O2	-O2	-O2

## 劈配 (mentor)

### 【题目背景】

一年一度的综艺节目《中国新代码》又开始了。Zayid 从小就梦想成为一名程序员，他觉得这是一个展示自己的舞台，于是他毫不犹豫地报名了。

### 【题目描述】

轻车熟路的 Zayid 顺利地通过了海选，接下来的环节是导师盲选，这一阶段的规则是这样的：

总共  $n$  名参赛选手（编号从 1 至  $n$ ）每人写出一份代码并介绍自己的梦想。接着由所有导师对这些选手进行排名。为了避免后续的麻烦，规定**不存在**排名并列的情况。

同时，每名选手都将**独立地**填写一份志愿表，来对总共  $m$  位导师（编号从 1 至  $m$ ）作出评价。志愿表上包含了共  $m$  档志愿。对于每一档志愿，选手被允许填写最多  $C$  位导师，每位导师**最多**被每位选手填写一次（放弃某些导师也是被允许的）。

在双方的工作都完成后，进行录取工作。每位导师都有自己战队的人数上限，这意味着可能有部分选手的较高志愿、甚至是全部志愿无法得到满足。

节目组对“前  $i$  名的录取结果最优”作出如下定义：

- 前 1 名的录取结果最优，当且仅当第 1 名被其最高非空志愿录取（特别地，如果第 1 名没有填写志愿表，那么该选手出局）。
- 前  $i$  名的录取结果最优，当且仅当在前  $i-1$  名的录取结果最优的情况下：第  $i$  名被其理论可能的最高志愿录取（特别地，如果第  $i$  名没有填写志愿表、或其所有志愿中的导师战队均已满员，那么该选手出局）。

如果一种方案满足“前  $n$  名的录取结果最优”，那么我们可以简称这种方案是最优的。

举例而言，2 位导师 T 老师、F 老师的战队人数上限分别都是 1 人；2 位选手 Zayid、DuckD 分列第 1、2 名。那么下面 3 种志愿表及其对应的最优录取结果如表中所示：

选手	第 1 志愿	第 2 志愿	录取志愿	加入战队
Zayid	N/A	T 老师、F 老师	2	F 老师
DuckD	T 老师	F 老师	1	T 老师

选手	第 1 志愿	第 2 志愿	录取志愿	加入战队
Zayid	T 老师	F 老师	1	T 老师
DuckD	T 老师	F 老师	2	F 老师

选手	第 1 志愿	第 2 志愿	录取志愿	加入战队
Zayid	F 老师	N/A	1	F 老师
DuckD	F 老师	N/A	出局	N/A

可以证明，对于上面的志愿表，对应的方案都是**唯一**的最优录取结果。

每个人都有自己的理想值  $s_i$ ，表示第  $i$  位同学希望自己被第  $s_i$  或更高的志愿录取，如果没有，那么他就会非常沮丧。

现在，所有选手的志愿表和排名都已公示。巧合的是，每位选手的排名都恰好与它们的编号相同。

对于**每一位**选手，Zayid 都想知道下面两个问题的答案：

- 在最优的录取方案中，他会被第几志愿录取。
- 在其他选手**相对排名**不变的情况下，至少上升多少名才能使得他不沮丧。

作为《中国新代码》的实力派代码手，Zayid 当然轻松地解决了这个问题。不过他还是想请你再算一遍，来检验自己计算的正确性。

### 【输入格式】

从文件 *mentor.in* 中读入数据。

每个测试点包含**多组测试数据**，第一行 2 个用空格隔开的非负整数  $T, C$ ，分别表示数据组数、每档志愿最多允许填写的导师数目。

接下来依次描述每组数据，对于每组数据：

- 第 1 行两个用空格隔开的正整数  $n, m$ 。
  - $n, m$  分别表示选手的数量、导师的数量。
- 第 2 行  $m$  个用空格隔开的正整数：其中第  $i$  个整数为  $b_i$ 。
  - $b_i$  表示编号为  $i$  的导师战队人数的上限。
- 第 3 行至第  $n + 2$  行，每行  $m$  个用空格隔开的非负整数：其中第  $i + 2$  行左起第  $j$  个数为  $a_{i,j}$ 。
  - $a_{i,j}$  表示编号为  $i$  的选手将编号为  $j$  的导师编排在了第  $a_{i,j}$  志愿。特别地，如果  $a_{i,j} = 0$ ，则表示该选手没有将该导师填入志愿表。
  - 在这一部分，保证每行中不存在某一个**正数**出现超过  $C$  次（0 可能出现超过  $C$  次），同时保证所有  $a_{i,j} \leq m$ 。
- 第  $n + 3$  行  $n$  个用空格隔开的正整数，其中第  $i$  个整数为  $s_i$ 。
  - $s_i$  表示编号为  $i$  的选手的理想值。
  - 在这一部分，保证  $s_i \leq m$ 。

**【输出格式】**

输出到文件 *mentor.out* 中。

按顺序输出每组数据的答案。对于每组数据，输出 2 行：

- 第 1 行输出  $n$  个用空格隔开的正整数，其中第  $i$  个整数的意义为：
  - 在最优的录取方案中，编号为  $i$  的选手会被该档志愿录取。
  - 特别地，如果该选手出局，则这个数为  $m + 1$ 。
- 第 2 行输出  $n$  个用空格隔开的非负整数，其中第  $i$  个整数的意义为：
  - 使编号为  $i$  的选手不沮丧，最少需要让他上升的排名数。
  - 特别地，如果该选手一定会沮丧，则这个数为  $i$ 。

**【样例 1 输入】**

```
3 5
2 2
1 1
2 2
1 2
1 1
2 2
1 1
1 2
1 2
2 1
2 2
1 1
0 1
0 1
2 2
```

**【样例 1 输出】**

```
2 1
1 0
1 2
0 1
1 3
0 1
```

**【样例 1 解释】**

三组数据分别与【题目描述】中的三个表格对应。

对于第 1 组数据：由于选手 1 没有填写第一志愿，所以他一定无法被第一志愿录取，也就一定会沮丧。选手 2 按原排名就不沮丧，因此他不需要提升排名。

对于第 2 组和第 3 组数据：1 号选手都不需要提升排名。而希望被第一志愿录取的 2 号选手都必须升到第 1 名才能如愿。

**【样例 2 输入】**

```
1 5
4 3
2 1 1
3 1 3
0 0 1
3 1 2
2 3 1
2 3 3 3
```

**【样例 2 输出】**

```
1 1 3 2
0 0 0 0
```

**【样例 2 解释】**

1 号选手的第一志愿只填写了 2 号导师，因此 1 号选手必定被 2 号导师录取。

2 号选手的第一志愿只填写了 3 号导师，因此 2 号选手必定被 3 号导师录取。

由于 2,3 号导师均满员，且 3,4 号选手均填写了 1 号导师，因此它们都会被 1 号导师录取。

所以 1,2 号选手均被第 1 志愿录取，3 号选手被第 3 志愿录取，4 号选手被第 2 志愿录取。

由于他们都如愿以偿了，所以他们都不需要提升名次。

**【样例 3】**

见选手目录下的 *mentor/mentor3.in* 与 *mentor/mentor3.ans*。

**【样例 4】**

见选手目录下的 *mentor/mentor4.in* 与 *mentor/mentor4.ans*。

**【子任务】**

测试点编号	$n \leq$	$m \leq$	$C$	其他约定
1	10	1	$= 1$	无
2		2	$= 2$	$s_i = m$
3		3	$= 3$	无
4	100	100	$= 1$	$b_i = 1$
5				无
6	200	200		$b_i = 1$
7				无
8	100	100	$= 10$	无
9	200	200		$b_i = 1$
10				无

对于所有测试点，保证  $T \leq 5$ 。

对于所有测试点中的所有数据，保证  $m \leq n \leq 200$ ， $b_i \leq n$ 。

## 林克卡特树 (lct)

### 【题目描述】

小 L 最近沉迷于塞尔达传说：荒野之息 (The Legend of Zelda: Breath of The Wild) 无法自拔，他尤其喜欢游戏中的迷你挑战。

游戏中有一个叫做“LCT”的挑战，它的规则是这样子的：现在有一个  $N$  个点的树 (Tree)，每条边有一个整数边权  $v_i$ ，若  $v_i \geq 0$ ，表示走这条边会获得  $v_i$  的收益；若  $v_i < 0$ ，则表示走这条边需要支付  $-v_i$  的过路费。小 L 需要控制主角 Link 切掉 (Cut) 树上的恰好  $K$  条边，然后再连接  $K$  条边权为 0 的边，得到一棵新的树。接着，他会选择树上的两个点  $p, q$ ，并沿着树上连接这两点的简单路径从  $p$  走到  $q$ ，并为经过的每条边支付过路费 / 获取相应收益。

海拉鲁大陆之神 TemporaryDO 想考验一下 Link。他告诉 Link，如果 Link 能切掉合适的边、选择合适的路径从而使 总收益-总过路费 最大化的话，就把传说中的大师之剑送给他。

小 L 想得到大师之剑，于是他找到了你来帮忙，请你告诉他，Link 能得到的 总收益-总过路费 最大是多少。

### 【输入格式】

从文件 *lct.in* 中读入数据。

输入第一行包含两个正整数  $N, K$ ，保证  $0 \leq K < N \leq 3 \times 10^5$ 。

接下来  $N - 1$  行，每行包含三个整数  $x_i, y_i, v_i$ ，表示第  $i$  条边连接图中的  $x_i, y_i$  两点，它的边权为  $v_i$ 。

### 【输出格式】

输出到文件 *lct.out* 中。

输出一行一个整数，表示答案。

### 【样例 1 输入】

```
5 1
1 2 3
2 3 5
2 4 -3
4 5 6
```

**【样例 1 输出】**

14

**【样例 1 解释】**

一种可能的最优方案为：切掉  $(2, 4, -3)$  这条边，连接  $(3, 4, 0)$  这条边，选择  $(p, q) = (1, 5)$ 。

**【样例 2】**

见选手目录下的 *lct/lct2.in* 与 *lct/lct2.ans*。

**【子任务】**

- 对于 10% 的数据， $k = 0$ ；
- 对于另外 10% 的数据， $k = 1$ ；
- 对于另外 15% 的数据， $k = 2$ ；
- 对于另外 25% 的数据， $k \leq 100$ ；
- 对于其他数据，没有特殊约定。

对于全部的测试数据，保证有  $1 \leq N \leq 3 \times 10^5, 1 \leq x_i, y_i \leq N, |v_i| \leq 10^6$ 。

**【提示】**

题目并不难。



## 制胡窜 (cutting)

### 【题目背景】

对于一个字符串  $S$ ，我们定义  $|S|$  表示  $S$  的长度。

接着，我们定义  $S_i$  表示  $S$  中第  $i$  个字符， $S_{L,R}$  表示由  $S$  中从左往右数，第  $L$  个字符到第  $R$  个字符依次连接形成的字符串。特别的，如果  $L > R$ ，或者  $L \notin [1, |S|]$ ，或者  $R \notin [1, |S|]$  我们可以认为  $S_{L,R}$  为空串。

### 【题目描述】

给定一个长度为  $n$  的仅由数字构成的字符串  $S$ ，现在有  $q$  次询问，第  $k$  次询问会给出  $S$  的一个字符串  $S_{l,r}$ ，请你求出有多少对  $(i, j)$ ，满足  $1 \leq i < j \leq n$ ， $i + 1 < j$ ，且  $S_{l,r}$  出现在  $S_{1,i}$  中或  $S_{i+1,j-1}$  中或  $S_{j,n}$  中。

### 【输入格式】

从文件 *cutting.in* 中读入数据。

输入的第一行包含两个整数  $n, q$ 。

第二行包含一个长度为  $n$  的仅由数字构成的字符串  $S$ 。

接下来  $q$  行，每行两个正整数  $l$  和  $r$ ，表示此次询问的子串是  $S_{l,r}$ 。

### 【输出格式】

输出到文件 *cutting.out* 中。

对于每个询问，输出一个整数表示合法的数对个数。

### 【样例 1 输入】

```
5 2
00100
1 2
1 3
```

### 【样例 1 输出】

```
5
1
```

**【子任务】**

测试点	$n$	$q$	其它约定
1	= 50	= 100	无
2 ~ 3	= 300	= 300	
4 ~ 5	= 2000	= 3000	
6 ~ 9	= 100000	= 100000	$\sum  S_{l,r}  \leq 10^6$
10 ~ 12	= 30000	= 50000	无
13	= 100000	= 100000	$S$ 中只有 0
14 ~ 20		= 300000	无

对于所有测试数据,  $1 \leq n \leq 10^5$ ,  $1 \leq q \leq 3 \cdot 10^5$ ,  $1 \leq l \leq r \leq n$ 。