

第二十三届全国青少年信息学奥林匹克联赛

CCF-NOIP-2017

提高组（复赛）第零试

竞赛时间：2017 年 11 月 10 日 8:30~12:00

题目名称	ZRX 切西瓜	ZRX 跑路	ZRX 的闪现
题目类型	传统型	传统型	传统型
目录	cut	Road	flash
可执行文件名	cut	Road	flash
输入文件名	cut.in	road.in	flash.in
输出文件名	cut.out	road.out	flash.out
每个测试点时限	1.0 秒	2.0 秒	2.0 秒
内存限制	256 MB	256 MB	256 MB
测试点数目	10	10	20
每个测试点分值	10	10	5

提交源程序文件名

对于 C++语言	cut.cpp	road.cpp	flash.cpp
对于 C 语言	cut.c	road.c	flash.c
对于 Pascal 语言	cut.pas	road.pas	flash.pas

编译选项

对于 C++语言	-lm	-lm	-lm
对于 C 语言	-lm	-lm	-lm
对于 Pascal 语言			

注意事项：

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. 除非特殊说明，结果比较方式均为忽略行末空格及文末回车的全文比较。
3. C/C++中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
4. 全国统一评测时采用的机器配置为：CPU AMD Athlon(tm) II x2 240 processor， 2.8GHz，内存 4G，上述时限以此配置为准。
5. 只提供 Linux 格式附加样例文件。
6. 评测在 NOI Linux 下进行。
7. 编译时不打开任何优化选项。

ZRX 切西瓜（cut）

【问题描述】

这么冷的天，ZRX 居然买了个西瓜回机房.....

可是这并不是普通的西瓜，所有西瓜籽在表面摆成了一个矩形点阵，共有 m 行，每行有 n 个西瓜籽。

ZRX 想要吃他的西瓜啦~

咔嚓一刀下去，恰好有 k 个西瓜籽被他的四十米长刀切成了两半。

因为 ZRX 是用左手切的瓜，所以刀切的方向一定是斜向右上方的，也就是斜率 $k \in (0, +\infty)$ 。

ZRXdalao 当然并不仅仅满足于吃瓜啦，他还想考考你.....

有多少种可能的切瓜方法呢？

【输入格式】

从文件 cut.in 中读入数据。

输入只有一行，分别为三个整数 n, m, k 。

整数之间由一个空格隔开。

【输出格式】

输出文件到 cut.out 中。

输出一个整数，表示切瓜的方法数。

【样例输入】

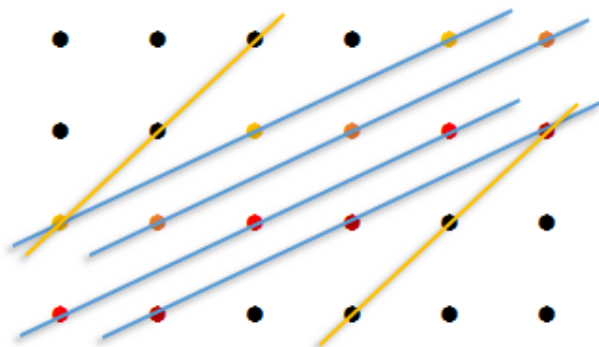
6 4 3

【样例输出】

6

【样例解释】

$n = 6, m = 4, k = 3$ 时，如图，共有 6 种斜向右上方的直线。



【数据范围】

对于 10%的数据， $n, m \leq 3$

对于 30%的数据， $n, m \leq 30$

对于 50%的数据， $n, m \leq 200$

对于 70%的数据， $n, m \leq 500$

对于 100%的数据， $n, m \leq 1000$

ZRX 跑路（road）

【问题描述】

ZRX 和他的小伙伴 ranwen、ranwan、renwen、ranven 想明天约着出去玩~

在这个城市中，共有 n 个地点（编号 $1 \dots n$ ）， m 条双向道路。走每条路会花费一定的时间。

ZRX 和小伙伴们一共有 p 个想要去的地方，记作 $f[i]$ 。

ZRX 和小伙伴们可以任意选定一个地方，作为明天的集合地点。然后明天他们会从这个点出发，经过所有的目的地 $f[i]$ ，然后结束一天的旅程。（不用回到起点）

路上的时间肯定是无聊的，所以他们想尽可能减少在路上花费的时间。

当然啦，如果确定了起点，机智的 ZRX 和 ranwen 肯定会走最优路线，但他们就是不会选出一个起点。

于是他们找到了你.....

所以你需要选出一个集合地点，使得在路上花费的时间最少。

【输入格式】

从文件 road.in 中读入数据。

输入的第一行包含三个正整数 n, m, p 。

接下来一行包含 p 个正整数 $f[i]$ ，表示目的地的编号。

接下来的 m 行，每行包含三个正整数 a, b, v ，表示 a 和 b 之间，有一条双向道路连接，通过这条路需要花费的时间为 v 。

【输出格式】

输出到文件 road.out 中。

输出一个正整数，表示你选择的起点的编号。

【样例输入】

```
6 12 3
3 6 2
6 3 1970
4 2 9830
1 4 9082
1 3 9162
2 5 4455
3 6 9454
3 1 4128
6 3 3996
```

3 2 8859

6 3 2975

1 6 193

6 4 4247

【样例输出】

2

【数据范围】

每个测试点的信息如下表：

测试点编号	n	m	p
1	10	30	3
2	100	3000	5
3	100	3000	8
4	80	1000	12
5	80	1000	13
6	1000	10000	8
7	100	1000	14
8	100	1000	15
9	1000	10000	15
10	1000	20000	15

并且，对于 100%的数据， $1 \leq f[i] \leq n$ ， $0 \leq v \leq 10000$

ZRX 的闪现 (flash)

【问题描述】

ZRX 最近迷上了一款叫“mizhi 闪现”的游戏。

游戏的地图是一棵树。

共 n 个节点（编号 $1 \dots n$ ），在每个节点处都有一个金币。根节点为 1 号点，每个节点都有一个点权 $w[i]$ 。

共 $n-1$ 条边，每条边都有一个时间值 $len[i]$ 。

所有的点之间都是连通的。

游戏开始的时候，时刻为 0，有一个 ZYT 站在 1 号点上。

你需要操控 ZYT，来吃掉所有的金币。

对于每一条边 i ，父节点到子节点需要花费时间 $len[i]$ ，而从子节点到父节点不花费时间（因为 ZYT 可以闪现 $= =$ ）。

ZYT 到达节点 i 处吃金币，需要消耗的能量 = 到达它的时刻 * 这个点的点权 $w[i]$ 。

ZYT 可不想花费它太多的能量，于是现在问你 ZYT 最少消耗多少能量。

【输入格式】

从文件 `flash.in` 中读入数据。

输入的第一行仅包含一个正整数 n 。

第二行包含 n 个正整数 $w[i]$ ，表示每个点的点权。

接下来的 $n-1$ 行中，每一行包含三个正整数 a, b, v ，表示 a 和 b 之间有一条边连接，时间值为 v 。（ a, b 不保证按照父节点和子节点的顺序给出）

保证输入数据合法。

【输出格式】

输出到文件 `flash.out` 中。

输出仅包含一个正整数，表示让 ZYT 吃到所有金币的最小的能量消耗。

【样例输入】

```
9
1 2 3 4 5 6 7 8 9
1 2 3
1 3 6
```

1 4 9

2 5 5

2 6 4

3 7 2

7 8 8

7 9 3

【样例输出】

840

【数据范围】

对于 20%的数据， $n \leq 20$ ，并且每个节点的儿子数 ≤ 3

对于 40%的数据， $n \leq 100$

对于 60%的数据， $n \leq 1000$

对于 80%的数据， $n \leq 10000$

对于 100%的数据， $n \leq 100000$ ， $0 \leq w[i]$ ， $len[i] \leq 100$