

## 第二十届全国青少年信息学奥林匹克竞赛 NOI2003

### 第二试

题目名称	数据生成器	草莓	智破连环阵
目录	day2/jerrygen	day2/berry	day2/zplhz
题目类型	普通	提交答案	普通
可执行文件名	jerrygen	-	zplhz
输入文件名	jerrygen.in	berry1.in~berry10.in	zplhz.in
输出文件名	jerrygen.out	berry1.out~berry10.out	zplhz.out
是否有部分分	否	是	是
题目总分	100	100	100
时间限制	2 秒	-	6 秒
内存限制	64M	-	64M

有关附加文件的信息，请参看具体的题目说明。

## 数据生成器

### 【题目背景】

小明在做 NOI2003 练习赛的《幸福的老鼠》时觉得题目太简单了，于是对原题做了一些扩展：

- 将原题的  $N$  从 20 扩展到 200000。
- 将原题经过一条街道需要的时间改为  $T_i$  ( $1 \leq T_i \leq 1000000000$ ) 分钟 ( $i$  为街道的编号)。
- 增加了一个条件：小狗家  $Y$  离老鼠家  $X$  的距离小于等于大狗家  $Z$  离老鼠家  $X$  的距离。

即使这样，他仍然很快地做了出来。于是，小明打算做一些输入文件来测试他的程序。现在他已经生成了一些符合题意的图，不过为了增大测试数据的难度，他希望你能帮他选取一组  $X$ 、 $Y$ 、 $Z$ ，使老鼠拿到礼物的时间尽可能地大。

### 【小明扩展的题目（注意，你并不需要解决此题）】

幸福的老鼠 Jerry 要过生日了，小狗大狗分别送了它一份生日礼物。现在 Jerry 打算从自己家  $X$  出发，先到小狗家  $Y$ （因为小狗家  $Y$  离老鼠家  $X$  的距离小于等于大狗家  $Z$  离老鼠家  $X$  的距离），再到大狗家  $Z$ ，将两份礼物取回。

卡通城由  $N$  ( $3 \leq N \leq 200000$ ) 个居住点和  $N-1$  条连接居住点的双向街道组成，经过第  $i$  条街道需花费  $T_i$  ( $1 \leq T_i \leq 1000000000$ ) 分钟的时间。可以保证，任两个居住点间都存在通路。

不妨设 Jerry 家在点  $X$ ，小狗家在点  $Y$ ，大狗家在点  $Z$ 。现在，请你计算，Jerry 最快需要耗费多长时间才能拿到生日礼物？

### 【任务描述】

定义：令  $|AB|$  表示卡通城中从  $A$  点走到  $B$  点需要的最少时间。

给出卡通城的地图，找到一组  $X$ 、 $Y$ 、 $Z$ ，使得：

$$|XY| \leq |XZ|$$

$$|XY| + |YZ| \text{ 最大。}$$

并求出此时  $|XY| + |YZ|$  的值。

### 【输入文件】

输入文件 jerrygen.in 第一行是两个整数  $N$  ( $3 \leq N \leq 200000$ ) 和  $M$  ( $M=N-1$ )，分别表示居住点总数和街道总数。从第 2 行开始到第  $N$  行，每行给出一条街道的信息。第  $i+1$  行包含整数  $U_i$ 、 $V_i$ 、 $T_i$  ( $1 \leq U_i, V_i \leq N, 1 \leq T_i \leq 1000000000$ )，表示街道  $i$  连接居住点  $U_i$  和  $V_i$ ，并且经过街道  $i$  需花费  $T_i$  分钟。

### 【输出文件】

输出文件 jerrygen.out 仅包含一个整数  $T$ ，即  $|XY| + |YZ|$  的最大值。

### 【样例输入】

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

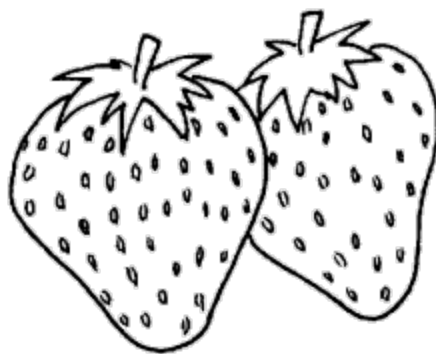
### 【样例输出】



## 草莓

### 【题目背景】

尽管不少人都吃过鲜美的草莓，却很少有人真正观察过野草莓的生长。它们从自己的枝上伸出一根根长长的触须，遇到合适的地方就会扎根发芽，长出一株新的草莓。所以，当你在森林中遇到一株草莓的时候，通常就意味着你会在它的周围找到一片草莓田。但这些草莓并非能够无忧无虑地生长，树林中穿梭的鸟儿和偶尔路过的鹿群都喜欢吃这种美味的浆果。不过，草莓最大的威胁却是来自那些贪吃的棕熊。他们不但可以吃掉整整一片草莓，而且还会粗鲁地把一片草莓田搞得乱七八糟。于是每当一块草莓田越长越大之后，森林中的精灵们就会把这片草莓田分成  $k$  块种到  $k$  个空地中去，以免被粗鲁的棕熊搞乱。她们希望每块空地上恰好放上一块用触须连接在一起的草莓田。不过，如果一块空地里的草莓太少，它们就会感到孤单，所以精灵们希望无论哪块空地含有草莓的总重量都不要太小。可是天真的精灵并不知道怎样来做这件事情，你可以帮助她们吗？



### 【任务描述】

定义： $sum_i$  表示第  $i$  块草莓田中所有草莓重量的和 ( $1 \leq i \leq k$ )。

$$x = \min \{ sum_i \mid 1 \leq i \leq k \}$$

你的任务就是要把一片草莓田分割成  $k$  块，且分割方案需要满足如下的条件：

- ☛ 每一块中的草莓必然是通过触须直接或者间接和其他草莓相连接的；
- ☛ 这种分割方案所对应的  $x$  尽可能的大。

最后输出你的分割方案和结果。

### 【输入说明】

第一行为三个整数  $n$ 、 $m$  及  $k$ ， $n$  表示草莓的株数， $m$  表示触须的数目， $k$  为空地的数目。

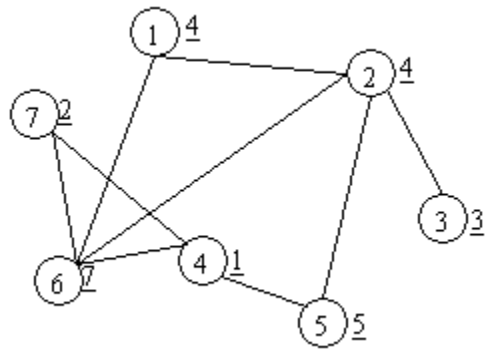
接下来的  $n$  行每行两个整数  $i$  及  $b_i$ ，表示第  $i$  株草莓的重量是  $b_i$  克。顺序下来的  $m$  行每行两个整数  $p$  和  $q$ ，表示第  $p$  株草莓和第  $q$  株草莓之间有一根触须相连接。

另外，在所有这些数据的最后还有单独的一行包括一个整数  $d$  用作评分系数，有关  $d$  的说明，可以参看下面的评分方法。

### 【输出说明】

你一共要输出  $k+1$  行。第一行为一个整数，表示你的分割方案中的  $x$ 。接下来的  $k$  行，每行表示一块草莓田。每行的第一个整数为  $n_i$ ，表示第  $i$  块田中的草莓株数。第二个到第  $n_i+1$  个整数为这些草莓的编号。请注意，这些草莓必然是通过触须相连接的。

【输入样例】



7 9 3  
1 4  
2 4  
3 3  
4 1  
5 5  
6 7  
7 2  
1 2  
1 6  
2 3  
2 5  
2 6  
4 5  
4 6  
4 7  
6 7  
2000000000

【输出样例】

7  
2 1 6  
2 2 3  
3 4 5 7

【评分方法】

本题是一道提交答案式题目，你需要针对给定的 10 个输入文件 2/berry1.in~2/berry10.in 提交你的输出文件 berry1.out~berry10.out（放在你的选手目录下）。我们将根据你提交的输出文件评分。对于某一确定的测试点来说，如果你的输出文件中第一行的  $x$

和下面的分割方案不符合，或者是输出文件本身就有错误，那么你将得不到该测试点的分数。这里输出文件的错误可以使用我们提供的 2/berry\_check 检查工具进行检查。只有当这个程序输出 Yes 的时候，你的输出才可以确认是可接受的。

对于可接受的输出，评分公式如下：

$$score = \left\lceil 10 \times e^{-\frac{2}{5} \times \left( \frac{d \times (best - x)}{best} \right)^2} \right\rceil$$

这里  $d$  为评分系数（输入数据中最后一行的整数）， $best$  为我们的最优结果。

**注意：可接受的输出不一定能够得分。**

### 【你如何测试自己的输出】

我们提供 2/berry\_check 这个工具来测试你的输出文件是否是可接受的。使用这个工具的方法是：

2/berry\_check <输入文件名> <输出文件名>

在你调用这个程序后，2/berry\_check 将根据你给出的输入和输出文件给出测试的结果，其中包括：

- ✖ 非法退出：未知错误；
- ✖ not connect：你程序输出的行中含有不连通的分量；
- ✖ duplicate：同一点输出了两次；
- ✖ extra：输出文件中包括多余数据；
- ✖ lack：输出的总点数不对；
- ✖ answer not match：输出中第一行的  $x$  和实际结果不符；
- ✔ Yes：输出可接受，将进行下一步的评分。

## 智破连环阵

### 【问题描述】

B 国在耗资百亿元之后终于研制出了新式武器——连环阵 (Zenith Protected Linked Hybrid Zone)，并声称这是一种无敌的自发性智能武器。但 A 国经侦察发现，连环阵其实是由  $M$  个独立武器组成的。这  $M$  个武器编号为  $1, 2, \dots, M$ 。每件武器有两种状态：无敌自卫状态和攻击状态。最初，1 号武器处于攻击状态，其他武器都处在无敌自卫状态。以后，一旦第  $i$  ( $1 \leq i < M$ ) 号武器被消灭，1 秒钟以后第  $i+1$  号武器就自动从无敌自卫状态变成攻击状态。当第  $M$  号武器被消灭以后，这个造价昂贵的连环阵就被彻底摧毁了。

为了打败 B 国，A 国军事部长打算用最廉价的武器——炸弹来消灭连环阵。经过长时间的精密探测，A 国的军事家们掌握了连环阵中  $M$  个武器的平面坐标，然后依此选择了  $n$  个点，并在这些点上安放了特殊的定时炸弹。这  $n$  个炸弹编号为  $1, 2, \dots, n$ 。每个炸弹的作用半径均为  $k$ ，且会持续爆炸 5 分钟。在这 5 分钟内，每枚炸弹都可以在瞬间消灭离它直线距离不超过  $k$  的、处在攻击状态的 B 国武器。和连环阵类似，最初  $a_1$  号炸弹持续引爆 5 分钟时间，然后  $a_2$  号炸弹持续引爆 5 分钟时间，接着  $a_3$  号炸弹引爆……以此类推，直到连环阵被摧毁。在每个炸弹爆炸的时候，其它尚未引爆的炸弹都处于地下隐蔽处，不会被己方的炸弹摧毁。

显然，选好  $a_1$ 、 $a_2$ 、 $a_3$ ... 十分重要。好的序列可以在仅使用较少炸弹的情况下就能将连环阵摧毁；坏的序列可能在使用完所有炸弹后仍无法将连环阵摧毁。现在，请你决定一个序列  $a_1$ 、 $a_2$ 、 $a_3$ ... 使得在第  $a_x$  号炸弹引爆的时间内连环阵被摧毁。这里的  $x$  应当尽量小。

### 【输入文件】

输入文件 `zplhz.in` 第一行包含三个整数： $M$ 、 $n$  和  $k$  ( $1 \leq M, n \leq 100$ ,  $1 \leq k \leq 1000$ )，分别表示 B 国连环阵由  $M$  个武器组成，A 国有  $n$  个炸弹可以使用，炸弹攻击范围为  $k$ 。以下  $M$  行，每行由一对整数  $x_i, y_i$  ( $0 \leq x_i, y_i \leq 10000$ ) 组成，表示第  $i$  ( $1 \leq i \leq M$ ) 号武器的平面坐标。再接下来  $n$  行，每行由一对整数  $u_i, v_i$  ( $0 \leq u_i, v_i \leq 10000$ ) 组成，表示第  $i$  ( $1 \leq i \leq n$ ) 号炸弹的平面坐标。输入数据保证无误和有解。

测试数据中的  $x_i$ 、 $y_i$ 、 $u_i$ 、 $v_i$  是随机生成的。

### 【输出文件】

输出文件 `zplhz.out` 的第一行包含一个整数  $x$ ，表示实际使用的炸弹数。第二行包括  $x$  个整数，依次表示  $a_1, a_2, \dots, a_x$ 。

### 【样例输入 1】

```
4 3 6
0 6
6 6
6 0
0 0
1 5
0 3
1 1
```

**【样例输出 1】**

2  
1 3

**【样例输入 2】**

10 10 45  
41 67  
34 0  
69 24  
78 58  
62 64  
5 45  
81 27  
61 91  
95 42  
27 36  
91 4  
2 53  
92 82  
21 16  
18 95  
47 26  
71 38  
69 12  
67 99  
35 94

**【样例输出 2】**

5  
6 2 1 3 4

**【评分标准】**

对每个测试点，如果你的输出合法，评分公式如下：



$$score = \begin{cases} 18, ans - good < -32 \\ 17, -32 \leq ans - good \leq -16 \\ 15, -15 \leq ans - good \leq -7 \\ 13, -6 \leq ans - good \leq -4 \\ 12, -3 \leq ans - good \leq -2 \\ 11, ans - good = -1 \\ 10, ans - good = 0 \\ 9, ans - good = 1 \\ 8, 2 \leq ans - good \leq 3 \\ 7, 4 \leq ans - good \leq 6 \\ 5, 7 \leq ans - good \leq 15 \\ 3, 16 \leq ans - good \leq 32 \\ 2, ans - good > 32 \end{cases}$$

其中  $good$  为我们的参考解， $ans$  为你的答案。

如果你的输出不合法，则该测试点只能得 0 分。

总分的计算公式：

$$\text{你这道题目的总分} = \min \left\{ 100, \sum_{i=1}^{10} score_i \right\}$$

其中  $score_i$  为你第  $i$  个测试点的得分。