

# NOI2020 模拟赛

--matthew99

每题的文件名为A/B/C.cpp，从A/B/C.in中读入，输出到A/B/C.out中。

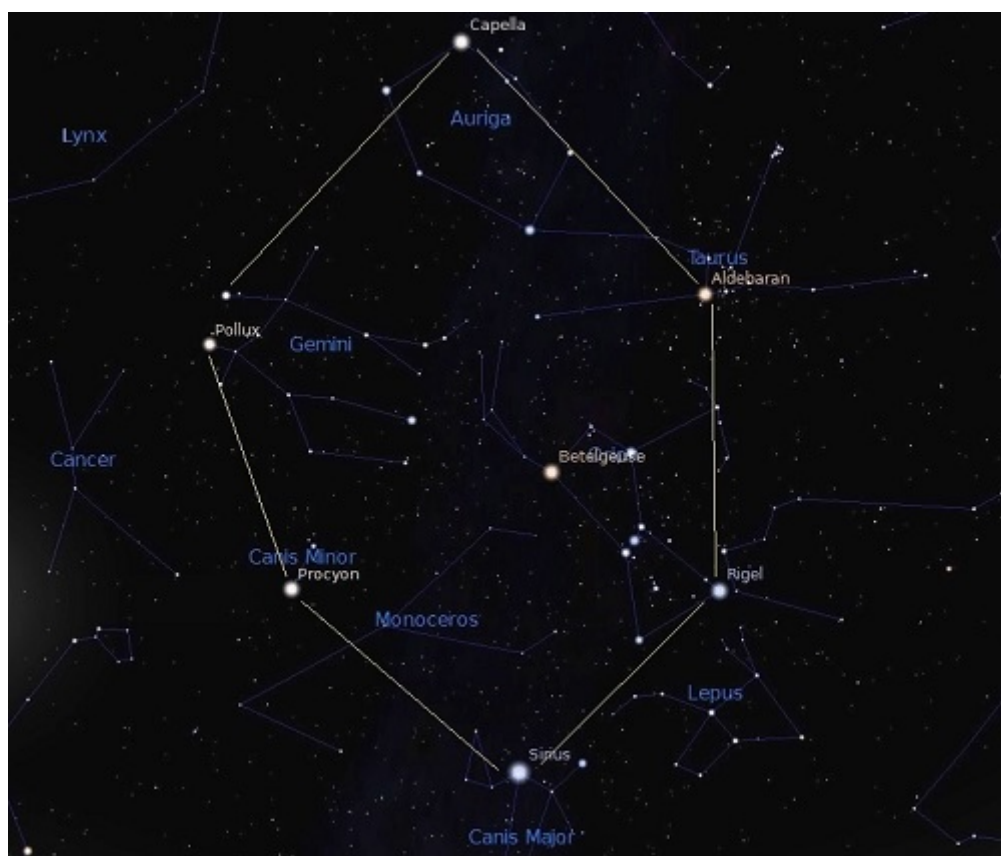
开启O2。

请至少开启C++11（C++14和17可酌情开启）。

## 观星(A)

### 问题描述

无法出门的你每天唯一能看到的是墙壁和窗外的星星，于是你对夜空起了兴趣。



小常识：上图为冬季六边形(Winter Hexagon)，由天狼星(Sirius)、参宿七(Rigel)、毕宿五(Aldebaran)、五车二(Capella)、北河三(Pollux)和南河三(Procyon)组成，而在中间是不久的将来将会发生超新星爆炸点亮夜空的参宿四(Betelgeuse)。作者今年（2020年）年初肉眼看到过这个六边形，图中的六边形虽小但是实际大小可以到整个夜空的一大半，你甚至无法一眼看到所有六颗星星。

你发现夜空可以看成是一个 $R^2$ 欧几里得坐标系，而每颗星星是坐标系上一个正整数点，没有三颗星共线，也没有两颗星重合。你发现了你最喜欢的一颗星，爱屋及乌，你从该星出发花了一个半径为 $R$ 的圆，并喜欢上了将圆内（包括边界）的点在内的所有星星。其他的星星，你都不喜欢。

你对着夜空玩起了消消看，你每次可以选择一颗你不喜欢的星星，和一颗你喜欢的星星，并将它们连线，然后两颗星星同时消去。

同时，天马行空的你又规定每一次选出的星星不能距离太远，它们连线的长度不能超过 $d$ 。

异想天开的你又规定，如果你选择的不喜欢的星星是 $u$ ，喜欢的星星是 $v$ ，那么不存在两颗你喜欢的且还没有被消去的星星 $a, b (a \neq v, b \neq v)$ ，使得 $a, b$ 距离 $u$ 也不超过 $d$ ，且 $u, v$ 的连线和 $a, b$ 的连线相交。

你希望消掉尽量多的星星，并希望求出一种方案。

## 输入格式

第一行三个正整数 $n, R, d$ 。

接下来 $n$ 行，每行两个正整数 $x_i, y_i$ ，表示第 $i$ 颗星星的坐标为 $(x_i, y_i)$ 。

## 输出格式

第一行一个数 $k$ ，表示你消去的星星的数目。显然， $k$ 为偶数。

接下来 $k \div 2$ 行，每行两个数 $u_i, v_i$ ，你选择的不喜欢的星星是 $u_i$ ，喜欢的星星是 $v_i$ ，请不要反向输出。

## 样例输入1

```
10 5530 5385
8 5730
5220 61
2896 2950
1025 649
5509 1773
6057 2432
6435 975
5366 8341
1127 3616
2849 1689
```

## 样例输出1

```
8
7 3
6 4
2 10
5 9
```

## 样例输入输出2

见下发文件。

## 说明

样例输出只是一种参考输出。最终测试时会使用特殊的测试器来检测你的输出是否正确。

## 测试器

## 数据规模和约定

对于5%的数据， $n \leq 10$ 。

对于10%的数据,  $n \leq 20$ 。

对于40%的数据,  $n \leq 100$ 。

对于60%的数据,  $n \leq 300$ 。

对于另外15%的数据,  $d = 2 \times 10^4$ 。

对于所有数据  $1 \leq n \leq 1,000, 1 \leq x_i, y_i \leq 10^4, 1, d, R \leq 2 \times 10^4$ 。没有三颗星共线, 也没有两颗星重合。

时限1s, 空间限制1G。

## 树的直径(B)

在一次编程竞赛课上, 你学习了树。老师说: “一棵树的直径就是这棵树上距离最远的两个节点之间的距离。现在给你们一棵树, 你们谁能求出它的直径?”。

你不假思索的说: “这不是傻逼题吗?”, 然后给出了一个十分优美的算法。

你的同学不服气, 说: “你的算法不够厉害!”, 然后给出了一个算法解决了带正权的情况。带正权的意思就是这棵树的边长不一定是 1, 而是可以为任意正数, 为了简化问题, 这里假设边权都是正整数。

这时候人群中站出来一个大神, 他说: “都别说话了, 我的算法更厉害, 可以解决这个问题一个更厉害的版本”。

接下来该大神给所有人介绍了他的加强版:

给定一个  $k$ , 你可以修改每条边的边权。修改的量可以为非整数, 但是每条边最终的边权不能为负。修改一条边有代价, 修改一条边的代价为这条最终的边权与初始边权的差的绝对值乘上这条边的单位代价。要求修改的总代价不超过一个给定值  $k$ 。你需要最小化最终树的直径并输出这个最小值。

听完这个版本, 全场除了你, 其他人都吓得目瞪口呆。然而你不服气, 并对大神说: “先别告诉我算法, 我自己想几分钟, 也可以想出来”。然而, 你想了很久都没有想出来。但是为了给自己一个台阶下, 你必须解决这个问题。

### 输入格式

本题包含多组数据。

输入的第一行包含一个整数  $T$  表示数据组数。

对于每组数据, 第一行包含一个整数  $n$  表示树的节点个数。

接下来  $n - 1$  行, 每行四个整数  $u_i, v_i, l_i, c_i$ 。表示节点  $u_i$  和  $v_i$  之间有一条边权为  $l_i$ , 单位代价为  $c_i$  的边。树上的节点从 1 开始编号。

### 输出格式

对于每组数据输出一行一个实数表示答案, 你的相对误差和绝对误差的最小值不得超过  $10^{-9}$ 。

### 样例输入1

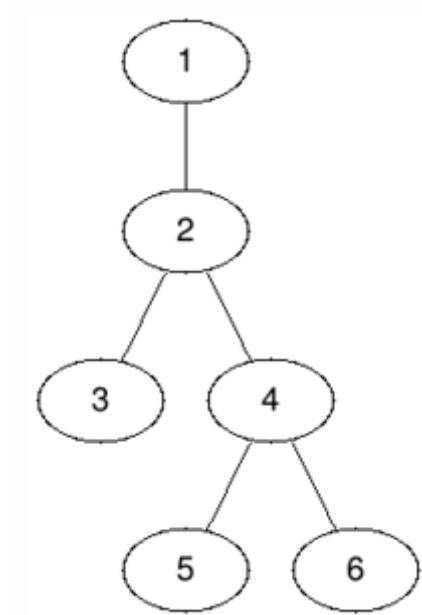
```
1
6 5
1 2 1 1
2 3 1 1
2 4 3 3
4 5 3 2
4 6 3 1
```

### 样例输出1

```
5
```

### 样例说明1

输入的树如图所示（该图使用<http://mxwell.github.io/draw-graph/?q=digraph> 生成）：



最优方案之一是将边 $(1, 2)$ ,  $(2, 3)$ ,  $(4, 5)$ ,  $(4, 6)$ 均减少1，代价为 $1 + 1 + 2 + 1 = 5$ 。这样之后树的直径之一是 $2 \rightarrow 4 \rightarrow 5$ ，长度为 $3 + (3 - 1) = 5$ 。

### 样例输入2

```
1
10 248
8 3 80 61
4 2 50 64
10 5 89 78
5 7 69 39
9 2 55 33
3 1 48 14
8 6 7 86
2 6 84 40
6 5 70 49
```

### 样例输出2

```
291.59183673
```

## 样例输入输出3

见下发文件

## 数据规模和约定

时间限制2s。

空间限制1G。

对于10%的数据， $n \leq 3$ 。

对于20%的数据， $n \leq 10$ 。

对于30%的数据， $n \leq 20$ 。

对于60%的数据， $n \leq 50$ 。

对于80%的数据， $n \leq 300$ 。

对于所有数据 $1 \leq T \leq 5, 1 \leq n \leq 2,000, 1 \leq l_i, c_i \leq 10^7, 1 \leq k \leq 10^{18}, 1 \leq u_i, v_i \leq n$ ，保证输入是一棵树。

## 线段树(C)

给定一个01序列，每次可以执行以下操作：

1. 操作一：给定两个等长不相交区间，将两个区间的第 $i$ 个位置同时修改为原来它们对应位置的数的二进制与运算的结果。
2. 操作二：将某一个位置取反。
3. 操作三：求一个区间内0或1的个数。

## 输入格式

第一行两个数 $n, m$ ，表示初始序列长度和操作数。

接下来一行一个长度为 $n$ 的01字符串，第 $i$ 个字符表示序列初始时第 $i$ 位的值。

接下来 $m$ 行，每行第一个数表示操作类型，1对应操作一，2和3对应操作二和三。

对于操作一，输入三个数 $l, r, \text{len}$ ，表示两个区间为 $[l, l + \text{len} - 1], [r, r + \text{len} - 1]$ ，保证区间不相交且不越界。

对于操作二，输入一个数 $x$ ，表示将序列的第 $x$ 个位置取反。

对于操作三，输入三个数 $l, r, t$ ， $t$ 为0或1，表示询问区间 $[l, r]$ 中 $t$ 的个数。

这里的区间 $[a, b]$ 定义为序列的第 $a$ 个位置到第 $b$ 个位置，且 $a \leq b$ 。

## 输出格式

对于操作三，输出一个数表示区间中对应的数的个数。

## 样例输入1

```
6 6
110101
1 4 1 2
2 4
3 4 5 0
2 4
1 4 3 1
3 3 6 1
```

### 样例输出1

```
2
1
```

### 样例输入输出2

见样例数据下载。

### 数据范围

对于30%的数据， $n, m \leq 30000$ 。

对于70%的数据， $m \leq 200000$ 。

对于100%的数据， $1 \leq m \leq 300000, 1 \leq n \leq 1000000$ 。

时间限制5s。

空间限制1G。