

# NOI2016 模拟赛

Ketsui Scape

2016.6.25

Task	Light	Vain	Death
Source file	light.cpp/pas/c	vain.cpp/pas/c	death.cpp/pas/c
Input file	light.in	vain.in	death.in
Output file	light.out	vain.out	death.out
Time Limit	2s	2s	2s
Memory Limit	512MiB	512MiB	512MiB
评测方式	Subtask	Subtask	Subtask
题目类型	传统型	传统型	传统型
是否有部分分	否	否	否
是否有附加文件	是	是	是

样例在下发文件夹的 Sample 文件夹内

因第二题和第三题读入量较大，建议使用快速读入。

在下发文件夹中有一份 fastIO.cpp 文件，C/C++ 选手可以选择使用里面的 fastIO 类来读入非负整数。

评测时开启 O2 指令

评测时系统栈大小为 64M

评测环境: Windows 10 Lemon v1.2

Processor : Intel® Core™ i7-6700K CPU @ 4.00GHz × 4

Memory : 15.7 GiB

保证在该评测环境下时限约为标程运行时间 4 倍

啊，小  $F$  睡着了呢。

# 1 Light

## 1.1 Description

梦境中  $n$  条光线从原点射出，和单位圆相交出  $n$  个点。

小  $F$  想从中选出  $k$  个点，使这  $k$  个点形成的凸包面积最大且原点被包含在凸包中。

请注意原点不能在凸包的边上。

## 1.2 Input

第一行，两个正整数  $n, k$ 。

接下来  $n$  行，每行一个浮点数  $Ang_i$  表示第  $i$  个点在单位圆上的角度。

角度以弧度制给出。

## 1.3 Output

输出一行  $k$  个整数，表示选择的点的标号。

若有多组解，输出任意一组。

若无解，输出  $-1$ 。

你的答案正确当且仅当与标准答案相差  $\leq 10^{-7}$ 。

## 1.4 Scoring

$$0 \leq Ang_i \leq 2\pi$$

Subtask 1 [10pts] :  $n \leq 20$

Subtask 2 [15pts] :  $n \leq 70$

Subtask 3 [25pts] :  $n \leq 200$

Subtask 4 [50pts] :  $n \leq 2000$

## 2 Vain

### 2.1 Description

小  $F$  来到了一片雪原上，雪原上坐落着  $n$  个小镇。

之前有一个大姐姐告诉小  $F$  她就在其中的一个小镇中。

“到时候来找我玩吧” 她们分别时是这么说的。

小镇之间有  $m$  条路，路结了冰，很难走，每条路有一个通行的困难值  $w_i$ 。

小  $F$  有第六感，她会凭她的感觉选定一个不喜欢的小镇  $i$ 。

选定之后，她可以选择任意除了  $i$  号小镇外的小镇作为起点，通过道路走访所有其他小镇，最后再走访到小镇  $i$ 。

小  $F$  现在很累，所以她希望她走到的道路的最大的通行困难值最小（这个值就是问题的答案）。

请注意小  $F$  可以走任意边任意多次。

### 2.2 Input

第一行两个正整数  $n, m$ 。

第二行  $m$  个正整数  $x_i, y_i, w_i$ ，表示一条双向道路连接了小镇  $x_i$  和小镇  $y_i$ ，通行困难值为  $w_i$ 。

### 2.3 Output

你要求出  $n$  个整数，第  $i$  个整数表示小  $F$  不喜欢的小镇的编号是  $i$  时的答案，若小镇  $i$  不可能是她到达的最后一个小镇输出  $-1$ 。

为了减小输出量，你只需要输出它们的和。

### 2.4 Scoring

$$1 \leq x_i, y_i \leq n, 1 \leq w_i \leq 10^9$$

保证无自环无重边。

Subtask 1 [10pts] :  $n, m \leq 2000$

Subtask 2 [30pts] :  $n, m \leq 100,000 \quad m \leq n + 60$

Subtask 3 [30pts] :  $n, m \leq 100,000$

Subtask 4 [30pts] :  $n, m \leq 1,000,000$

## 2.5 Hint

简述题意，对于  $1 \leq i \leq n$ ，求出点  $i$  度数强制为 1 的情况下最小生成树的最大边的权值。

对于样例 *vain0.in*

当  $i = 1$  时，在 MST 上的边有 2, 4, 5, 6 号边，答案为 4。

当  $i = 2$  时，在 MST 上的边有 1, 2, 5, 6 号边，答案为 4。

当  $i = 3$  时，不存在这样的 MST，答案为 -1。

当  $i = 4$  时，在 MST 上的边有 1, 2, 4, 6 号边，答案为 3。

当  $i = 5$  时，在 MST 上的边有 1, 2, 4, 6 号边，答案为 3。

故应输出  $4 + 4 - 1 + 3 + 3 = 13$

## 3 Death

### 3.1 Description

小  $F$  堕入了梦魇的世界，一只巨大的魔爪，或者说，一棵根节点为 1 的有根树，每个节点有一个危险值  $w_i$ 。

小  $F$  不会自己从梦境中醒来，她要在梦境中切断自己与梦魇的联系，才能逃离这个世界。

每次，小  $F$  都是从 1 号节点进入这个世界。

她要去编号为  $x_1, x_2 \dots x_k$  的节点上切断联系。

她希望知道她要经过的所有节点的危险度之和最小是多少（多次经过只算一次）。

### 3.2 Input

第一行两个正整数  $n, m$ ，树的节点数和总操作数。

第二行  $n$  个整数  $w_i$ ，表示  $n$  个节点的危险度。

第三行， $n - 1$  个整数  $f_i$  表示节点  $i + 1$  的父节点。

接下来的  $m$  行，每行表示一个操作。

1.  $1\ x\ w$  在编号为  $x$  的点下加一个危险值为  $w$  的叶节点。

2.  $2\ x\ w$  令  $fa_x$  为此刻  $x$  的父节点标号，在边  $(fa_x, x)$  上新建一个危险值为  $w$  的节点  $z$ ，原来的边变为  $(fa_x, z)$  和  $(z, x)$

3.  $3\ k\ x_1\ x_2 \dots x_k$  表示一次询问，意义如题面所示。

新建的节点编号为当前尚未使用的最小正整数。

强制在线，对于所有操作中涉及的节点编号，令读入的节点编号为  $x$ ，则真实的编号为  $(x + lastans) \bmod cnt + 1$ 。其中  $lastans$  为上一次询问的答案， $cnt$  为操作前的节点个数，初始  $lastans$  为 0。

### 3.3 Output

对于每个询问，输出一行整数，表示答案。

### 3.4 Scoring

修改和询问中所提到的节点编号在解密后均保证存在。

$$1 \leq w \leq 10^9$$

Subtask 1 [10pts] :  $n, m, \sum k \leq 2000$

Subtask 2 [20pts] :  $n, m, \sum k \leq 200,000$  保证树随机生成

Subtask 3 [30pts] :  $n, m, \sum k \leq 200,000$     保证不存在第二种操作

Subtask 4 [40pts] :  $n, m, \sum k \leq 200,000$