

快乐的打码【3.30】

简介：

又是一个全新模式的打码哦

下面是一些注意事项：

- 1、时间：下午 2.半到 6. 可以提前交卷
- 2、题数：4 题，满分 400 分
- 3、赛制：采用 NOIp 普及组的 OI 赛制，每道题有且只有一次提交机会，交卷后才公布成绩
- 4、提示：部分分很足，部分题目写简单的程序可以混不少分
- 5、210+分 = 普及组一等奖 = 优秀的初中生水平

题目来源：

- 1、NOIp 普及组 2015 T1（入门）
- 2、普及赛模拟题（普及-）
- 3、NOIp 普及组 2018 T3（普及+/提高）
- 4、NOI 2015 决赛 Day1 T1（提高+/省选-）

题目：

1、Gold（入门）

题目描述 展开

国王将金币作为工资，发放给忠诚的骑士。第一天，骑士收到一枚金币；之后两天（第二天和第三天），每天收到两枚金币；之后三天（第四、五、六天），每天收到三枚金币；之后四天（第七、八、九、十天），每天收到四枚金币.....；这种工资发放模式会一直这样延续下去：当连续N天每天收到N枚金币后，骑士会在之后的连续 $N + 1$ 天里，每天收到 $N + 1$ 枚金币。

请计算在前 K 天里，骑士一共获得了多少金币。

输入格式

一个正整数 K ，表示发放金币的天数。

输出格式

一个正整数，即骑士收到的金币数。

输入输出样例

输入 #1	复制	输出 #1	复制
6		14	
输入 #2	复制	输出 #2	复制
1000		29820	

说明/提示

【输入输出样例 1 说明】

骑士第一天收到一枚金币；第二天和第三天，每天收到两枚金币；第四、五、六天，每天收到三枚金币。因此一共收到 $1 + 2 + 2 + 3 + 3 + 3 = 14$ 枚金币。

对于 100% 的数据， $1 \leq K \leq 10,000$ 。

2、Minions Have Spawned（普及-）

题目描述

目前，游戏共有 31 个英雄，每局可以派 6 个英雄上场。实际能使用的英雄数将会小于 31。简单地假设，每位英雄对团队的贡献可以用一个正整数 v_i 表示。团队的总能力等于所有英雄的贡献之和。特别地，对于 m 组特定的英雄，他们一起出现会让团队总能力提高 x_i 。

另注：英雄组合可以重复，若重复请多次计数。

输入格式

第一行两个整数 n, m ，表示可选英雄个数和特定的组合数。

接下来 n 行，每行先给出第 i 个英雄的名字，然后再输入一个整数 v_i 。

接下来 m 行，每行两个字符串 a, b 和一个整数 x_i ，表示若英雄 a 和英雄 b 一起出现时总能力提高 x_i 。

保证： a 和 b 不是同一个英雄。

输出格式

一个整数，表示团队能力值的最大值。

输入输出样例

输入 #1

复制

```
7 0
Sigma 10
Orisa 10
Mei 10
Reaper 10
Moirra 10
Lucio 10
Doomfist 1
```

输出 #1

复制

60

输入 #2

复制

```
8 2
Pharah 5
Mercy 5
Reinhardt 10
DVA 10
SOLDIER:76 12
Torbjorn 5
Anna 8
Baptiste 10
Pharah Mercy 20
Anna Pharah 5
```

输出 #2

复制

75

说明/提示

样例 1 解释

我们应该选择前 6 名英雄。

样例 2 解释

我们应该选择 Pharah、Mercy、Reinhardt、DVA、SOLDIER:76、Anna

数据范围

对于 40% 的数据, $m = 0$ 。

对于 100% 的数据, $6 \leq n \leq 30$, $0 \leq m \leq 30$, $1 \leq x_i, v_i \leq 100$, 英雄名字长度 ≤ 10 , 字符集包括大小写字母、数字以及冒号 (:)。

3、Symmetric Binary Tree (普及+/提高)

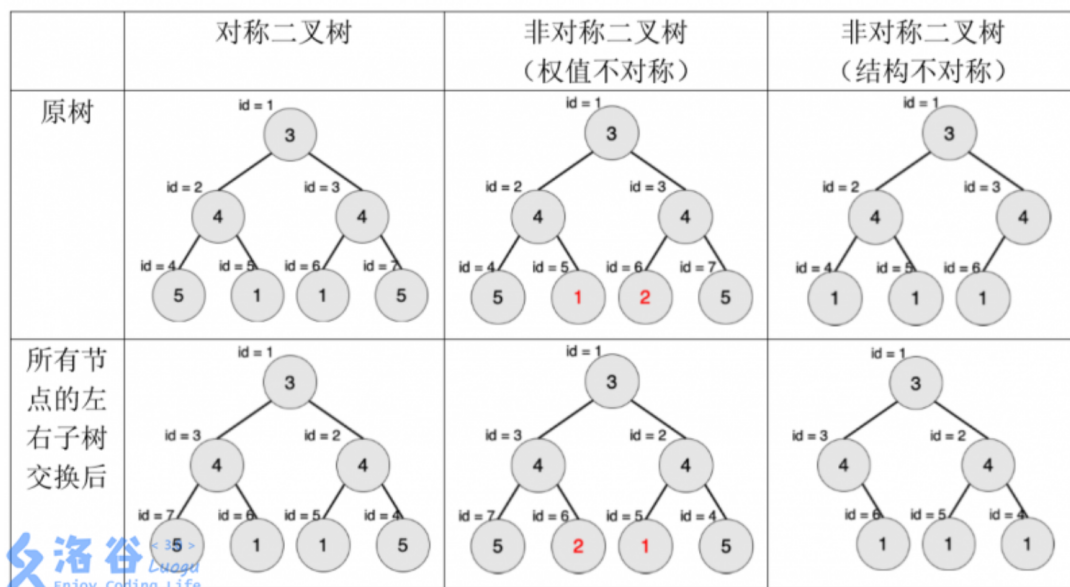
题目描述

[\[\] 展开](#)

一棵有点权的有根树如果满足以下条件, 则被轩轩称为对称二叉树:

1. 二叉树;
2. 将这棵树所有节点的左右子树交换, 新树和原树对应位置的结构相同且点权相等。

下图中节点内的数字为权值, 节点外的 id 表示节点编号。



现在给出一棵二叉树, 希望你找出它的一棵子树, 该子树为对称二叉树, 且节点数最多。请输出这棵子树的节点数。

注意: 只有树根的树也是对称二叉树。本题中约定, 以节点 T 为子树根的一棵“子树”指的是: 节点 T 和它的全部后代节点构成的二叉树。

输入格式

第一行一个正整数 n ，表示给定的树的节点的数目，规定节点编号 $1 \sim n$ ，其中节点 1 是树根。

第二行 n 个正整数，用一个空格分隔，第 i 个正整数 v_i 代表节点 i 的权值。

接下来 n 行，每行两个正整数 l_i, r_i ，分别表示节点 i 的左右孩子的编号。如果不存在左 / 右孩子，则以 -1 表示。两个数之间用一个空格隔开。

输出格式

输出文件共一行，包含一个整数，表示给定的树的最大对称二叉子树的节点数。

输入输出样例

输入 #1

[复制](#)

输出 #1

[复制](#)

```
2
1 3
2 -1
-1 -1
```

```
1
```

输入 #2

[复制](#)

输出 #2

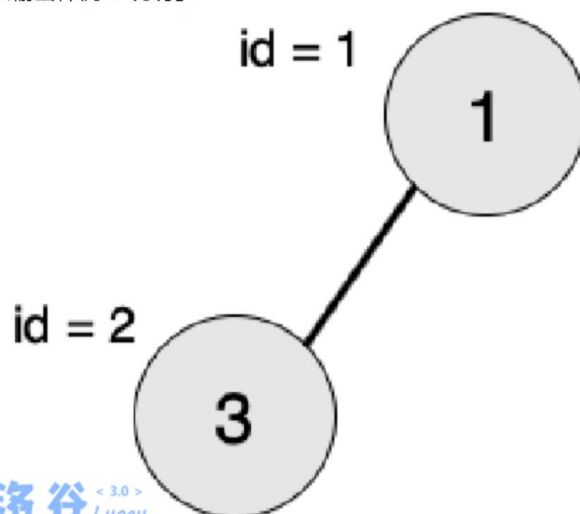
[复制](#)

```
10
2 2 5 5 5 5 4 4 2 3
9 10
-1 -1
-1 -1
-1 -1
-1 -1
-1 2
3 4
5 6
-1 -1
7 8
```

```
3
```

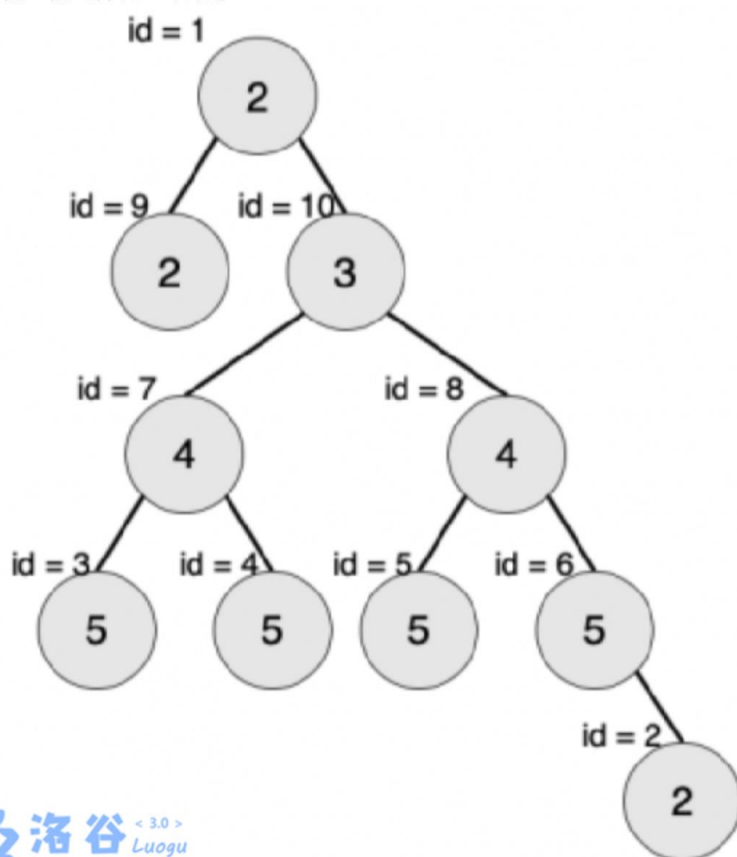
说明/提示

【输入输出样例 1 说明】



最大的对称二叉子树为以节点 2 为树根的子树，节点数为 1。

【输入输出样例 2 说明】



最大的对称二叉子树为以节点 7 为树根的子树，节点数为 3。

【数据规模与约定】

共 25 个测试点。

$v_i \leq 1000$ 。

测试点 1 ~ 3, $n \leq 10$, 保证根结点的左子树的所有节点都没有右孩子, 根结点的右子树的所有节点都没有左孩子。

测试点 4 ~ 8, $n \leq 10$ 。

测试点 9 ~ 12, $n \leq 10^5$, 保证输入是一棵“满二叉树”。

测试点 13 ~ 16, $n \leq 10^5$, 保证输入是一棵“完全二叉树”。

测试点 17 ~ 20, $n \leq 10^5$, 保证输入的树的点权均为 1。

测试点 21 ~ 25, $n \leq 10^6$ 。

本题约定：

层次：节点的层次从根开始定义起，根为第一层，根的孩子为第二层。树中任一节点的层次等于其父亲节点的层次加 1。

树的深度：树中节点的最大层次称为树的深度。

满二叉树：设二叉树的深度为 h ，且二叉树有 $2^h - 1$ 个节点，这就是满二叉树。

完全二叉树：设二叉树的深度为 h ，除第 h 层外，其它各层的结点数都达到最大个数，第 h 层所有的结点都连续集中在最左边，这就是完全二叉树。

4、Program Automatic Analysis（提高+/省选-）

题目描述

[展开](#)

在实现程序自动分析的过程中，常常需要判定一些约束条件是否能被同时满足。

考虑一个约束满足问题的简化版本：假设 x_1, x_2, x_3, \dots 代表程序中出现的变量，给定 n 个形如 $x_i = x_j$ 或 $x_i \neq x_j$ 的变量相等/不等的约束条件，请判定是否可以分别为每一个变量赋予恰当的值，使得上述所有约束条件同时被满足。例如，一个问题中的约束条件为： $x_1 = x_2, x_2 = x_3, x_3 = x_4, x_4 \neq x_1$ ，这些约束条件显然是不可能同时被满足的，因此这个问题应判定为不可被满足。

现在给出一些约束满足问题，请分别对它们进行判定。

输入格式

输入的第一行包含一个正整数 t ，表示需要判定的问题个数。注意这些问题之间是相互独立的。

对于每个问题，包含若干行：

第一行包含一个正整数 n ，表示该问题中需要被满足的约束条件个数。接下来 n 行，每行包括三个整数 i, j, e ，描述一个相等/不等的约束条件，相邻整数之间用单个空格隔开。若 $e = 1$ ，则该约束条件为 $x_i = x_j$ 。若 $e = 0$ ，则该约束条件为 $x_i \neq x_j$ 。

输出格式

输出包括 t 行。

输出文件的第 k 行输出一个字符串 YES 或者 NO（字母全部大写），YES 表示输入中的第 k 个问题判定为可以被满足，NO 表示不可被满足。

输入输出样例

输入 #1

[复制](#)

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

输出 #1

[复制](#)

```
NO
YES
```

输入 #2

[复制](#)

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

输出 #2

[复制](#)

```
YES
NO
```

说明/提示

【样例解释1】

在第一个问题中，约束条件为： $x_1 = x_2, x_1 \neq x_2$ 。这两个约束条件互相矛盾，因此不可被同时满足。

在第二个问题中，约束条件为： $x_1 = x_2, x_1 = x_2$ 。这两个约束条件是等价的，可以被同时满足。

【样例说明2】

在第一个问题中，约束条件有三个： $x_1 = x_2, x_2 = x_3, x_3 = x_1$ 。只需赋值使得 $x_1 = x_2 = x_3$ ，即可同时满足所有的约束条件。

在第二个问题中，约束条件有四个： $x_1 = x_2, x_2 = x_3, x_3 = x_4, x_4 \neq x_1$ 。由前三个约束条件可以推出 $x_1 = x_2 = x_3 = x_4$ ，然而最后一个约束条件却要求 $x_1 \neq x_4$ ，因此不可被满足。

【数据范围】

所有测试数据的范围和特点如下表所示

测试点编号	n 的规模	i, j 的规模	约定
1	$1 \leq n \leq 10$	$1 \leq i, j \leq 10,000$	$1 \leq t \leq 10$ $e \in \{0,1\}$
2			
3			
4			
5	$1 \leq n \leq 100$		
6			
7			
8	$1 \leq n \leq 100,000$		
9			
10			

注：实际上 $n \leq 10^6$ 。