

# 冲刺 NOIP2023 模拟试题

8 月 22 日

时间：7:50-12:20

(请选手务必仔细阅读本页内容)

## 一. 题目概况

题目名称	数据	路哥	质数	组合
题目类型	传统型	传统型	传统型	传统型
目录	data	luge	prime	combo
可执行文件名	data	luge	prime	combo
输入文件名	data.in	luge.in	prime.in	combo.in
输出文件名	data.out	luge.out	prime.out	combo.out
每个测试点时限	2.0 秒	1.0 秒	2.0 秒	1.0 秒
内存限制	512 MB	256 MB	512 MB	256 MB
测试点数目	20	20	20	20
每个测试点分值	5	5	5	5

## 二. 提交源程序文件名

对于 C++ 语言	data.cpp	luge.cpp	combo.cpp	combo.cpp
-----------	----------	----------	-----------	-----------

## 三. 编译选项

对于 C++ 语言	-lm -std=c++14 -O2
-----------	--------------------

## 四. 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、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。
- 3、全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz，内存 32GB。上述时限以此配置为准。
- 4、只提供 Linux 格式附加样例文件。
- 5、特别提醒：评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以其为准。

## 1. 数据

(data)

### 【题目背景】

宫水三叶被要求出一场测试，于是三叶非常认真的出了一套题。

然而在临近测试时，一场意外使得这场测试所有的资料全部被清空了。其中包括题面，题解，标程和数据。

无奈之下，三叶找到了小 H，希望她能恢复这些被清空的数据。

### 【题目描述】

擅长电脑的小 H 明白，这些数据是可恢复的，所以小 H 决定帮助三叶。

由于某些原因，这些数据有依赖关系，复原一个数据需要复原它所依赖的数据。这些依赖关系形成一棵树。

对于一个数据，假设它有两个系数  $a_i, b_i$ ，在复原这个数据时，因为还要保证没有被复原的数据的完整性，需要的代价为：

$$b_i \cdot \sum_{j \text{ 还没被复原}} a_j$$

小 H 认为自己一定可以在规定时间内复原出数据，但是她想捉弄一下三叶。因此小 H 想要让复原的时间尽可能长，也就是让上述代价尽量大。

现在你知道  $n$  个数据的系数和它们之间的依赖关系，请求出最大代价。

### 【形式化题意】

给定一棵  $n$  个点的树，对于所有  $2 \leq i \leq n$ ，它的父亲节点为  $f_i$ ，每一个点有两个系数  $a_i, b_i$ 。

你需要求出一个长度为  $n$  的排列，满足对于  $2 \leq i \leq n$ ， $f_i$  都在  $i$  出现前出现。

这个排列的代价为：

$$\sum_{i=1}^n \left( b_{p_i} \cdot \sum_{j=i+1}^n a_{p_j} \right)$$

求最大代价。

### 【输入格式】

从文件 `data.in` 中读入数据。

第一行一个整数  $n$ ，表示需要复原的数据的个数。

第二行  $n-1$  个整数，第  $i$  个数表示  $f_{i+1}$ ，即复原第  $i+1$  个数据需要先复原第  $f_{i+1}$  个数据。

接下来  $n$  行，每行两个整数  $a_i, b_i$ ，表示第  $i$  个数据自身的两个系数。

### 【输出格式】

输出到文件 data.out 中。

输出一行，表示最大的代价。

### 【样例输入】

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

### 【样例输出】

26

### 【数据范围与约定】

测试点	n<=	特殊性质
1	20	-
2		
3		
4	1000	
5		
6		
7	5*10^4	
8		
9		
10	3*10^5	fi=i-1
11		fi=1
12		
13		-
14		
15		
16		
17		
18		
19		
20		

对于所有数据，满足  $1 \leq n \leq 3 \cdot 10^5$ ， $1 \leq a_i, b_i \leq 5000$ ， $1 \leq f_i < i$ 。

## 2. 路哥

(luge)

### 【问题描述】

众所周知，曾经的杨吞天是个霸气威风的大男孩，他心中一直住着一位女生。

9080 年，年迈的杨吞天突然想起了自己当年为了追求 Luge 所作出种种傻事，他从自己的杂物柜中翻出了一张草稿纸，上面密密麻麻的点竟然构成了一棵树，他仔细回想，终于想起了那段往事。

当年，Luge 是一位潇洒的奇女子，她的身上常常有着各式各样的花朵，她为了响应杨吞天的追求，对他承诺：只要他能在小镇中捡到留下的  $k$  朵花，她就嫁给杨吞天。杨吞天为了实现自己日思夜想的梦想，特地在草稿纸上画出了小镇的模样以更好地安排自己的路线。他的捡花规则如下：

- 小镇的形态是一棵树，每个结点处都有一些花朵。xx 学校的位置在 1 号点，杨吞天会从这里出发去捡花朵，不断地沿着树边走，直到将他能走到的结点的**所有花朵都捡起来**；
- 杨吞天为了更大概率地追求 Luge，他在测试的前一天去请教了一位懂得仙术的大神青蛙人，青蛙人说：“爱情的事情要靠缘分，我随机地帮你断掉一些道路，这样可以避免你鬼迷心窍将所有花朵都捡了。”于是青蛙人会随机断掉一些边（当然有可能一条边都不断，每条边被断掉的概率为 0.5 且相互独立）将杨吞天的路线破坏掉。

但是就当他信心满满地准备参加测试的时候，他却去参加 IOI 了！现在他想起了这件事来找到了你，你作为一位大学者，被杨吞天强迫着打破爱情的禁忌告诉他当年成功追求到 Luge 的概率。

### 【输入格式】

第一行两个正整数  $n, k$ ，分别表示小镇的结点个数和 Luge 承诺的花朵数。

接下来一行  $n$  个非负整数，第  $i$  个数表示第  $i$  号结点的花朵数。

接下来  $(n - 1)$  行，每行两个正整数表示一条树边，描述小镇的连通状况。

### 【输出格式】

输出一行一个非负整数表示杨吞天追求到 Luge 的概率，答案对 998244353 取模。

### 【样例输入 1】

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

### 【样例输出 1】

```
499122177
```

**【输入输出样例 1 说明】**

断掉第 1, 2 条边或者只断第 1 条边均可恰好获得 1 朵花朵, 这两种情况的概率的和为 0.5。

**【样例输入 2】**

```
9 4
0 0 1 4 1 2 0 4 7
1 2
2 5
3 1
3 4
2 6
4 8
3 7
8 9
```

**【样例输出 2】**

```
967049217
```

**【数据范围与约定】**

- 对于 20% 的数据, 满足  $n \leq 20$ ,  $k \leq 100$ ;
- 对于 40% 的数据, 满足  $n, k \leq 300$ ;
- 对于 60% 的数据, 满足  $n, k \leq 1000$ ;
- 另有 10% 的数据, 满足树的每个结点最多只有两条树边与之相连。

对于 100% 的数据, 满足  $1 \leq n, k \leq 5000$ , 树的每个结点处的花朵个数不超过 5000。

### 3. 质数

(prime)

#### 【问题描述】

小  $\omega$  是个可爱的女孩子，她特别喜欢质数。

现在她有  $n$  个复数，分别是  $a_i$  ( $1 \leq i \leq n$ )。

一个数是质数当且仅当它是大于 1 的正整数而且它没有 1 和自己外的正因子。

现在她想让你支持几个操作：

1. 给定  $l, r, x, y$ ，表示将  $a_l$  至  $a_r$  改为  $x + yi$ 。其中  $i$  是虚数单位， $i^2 = -1$ 。

2. 给定  $l, r, x, y$ ，表示将  $a_l$  至  $a_r$  乘上  $x + yi$ 。其中  $i$  是虚数单位， $i^2 = -1$ 。

3. 给定  $l, r$ ，询问  $a_l$  至  $a_r$  有几个质数。

可以发现的是： $(a_0 + b_0i) \times (a_1 + b_1i) = a_0 \times a_1 - b_0 \times b_1 + (a_0 \times b_1 + a_1 \times b_0)i$ 。

#### 【输入格式】

第一行两个正整数  $n, q$ ，表示数的个数和操作数。

下面  $n$  行，每行有两个整数  $u_i, v_i$ ，表示  $a_i = u_i + v_i i$ 。

下面  $q$  行，每行第一个正整数  $opt_i$  表示操作编号。

若  $opt_i = 1, 2$ ，那么下面紧跟四个整数  $l_i, r_i, x_i, y_i$ ，表示进行一次对应操作。

若  $opt_i = 3$ ，那么下面紧跟两个整数  $l_i, r_i$ ，表示进行一次询问。

#### 【输出格式】

对于每个 3 操作，输出一行一个非负整数，表示询问的答案。

#### 【样例输入 1】

```
7 7
0 -2
1 1
1 2
2 0
-2 0
-1 2
0 2
3 2 4
3 3 6
1 5 7 2 -1
1 4 5 2 -1
3 1 4
1 5 5 -2 2
3 3 3
```

#### 【样例输出 1】

```
1
1
0
0
```

### 【数据范围】

对于所有的测试点，保证  $1 \leq n, q \leq 2 \times 10^5$ ;  $-5000 \leq x_i, y_i, u_i, v_i \leq 5000$ 。

测试点	特殊条件
1	保证 $1 \leq n, m \leq 5000$
2	
3	
4	
5	
6	
7	保证 $v_i = y_i = 0$
8	
9	
10	
11	
12	
13	-
14	
15	
16	
17	
18	
19	
20	



## 4. 组合

(combo)

### 【问题描述】

我是小 j，我有一道组合题，和一棵树  $G(V, E)$ 。

你要在图  $G$  上玩一个操作游戏，每次操作你先置  $G' = G$ ，然后选择一个顶点  $v \in V$ ，给  $G'$  加入一个顶点  $v'$ ；之后你要对于  $\forall u \in V, (u, v) \in E$ ，给  $G'$  加入一条边  $(u, v')$ ；最后把  $G$  替换为  $G'$ ，开始下一轮操作。

当然你一直这样操作下去也没有意思。所以某次操作开始前，若  $G$  存在一条**哈密顿路**，我就会把游戏终止。哈密顿路呢就是说从任意点出发、不重复地经过所有点、以任意点结束的一条简单路径。

现在我要你求出最少的操作次数及对应的步骤，使得游戏终止。众所周知判断图是否存在哈密顿路是 NPC 问题，所以你得报告一条哈密顿路。

### 【输入格式】

第一行一个整数  $n$ ，那么  $G$  的顶点集初始为  $\{V_1, V_2, \dots, V_n\}$ 。

下面  $n - 1$  行，每行两个整数  $(a, b)$ ，表示  $(V_a, V_b) \in E$ 。

### 【输出格式】

第一行一个整数，表示所求的最少操作次数  $g$ 。 $g$  是一个有限的数，我可以明确地告诉你这一点。

第二行  $g$  个整数  $u_1, u_2, \dots, u_g$ ，那么第  $i$  次操作是选择  $v = V_{u_i}, v' = V_{n+i}$ ，容易看出  $1 \leq u_i \leq n+i-1$ 。

最后一行  $n + g$  个整数  $p_1, p_2, \dots, p_{n+g}$ ，表示终止时发现了一条哈密顿路  $V_{p_1} \rightarrow V_{p_2} \rightarrow \dots \rightarrow V_{p_{n+g}}$ 。

### 【样例输入】

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

### 【样例输出】

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

### 【数据范围与约定】

对所有数据，保证  $3 \leq n \leq 100$ 。

对于每个数据点，如果你的答案只有  $g$  正确，那么你可以获得 60% 的分数，但注意这种情况下你需要输出一个**格式正确**的答案，即输出了数量正确的数且满足  $1 \leq u_i \leq n + i - 1, 1 \leq p_i \leq n + g$ ，否则会被判为 0 分；如果你的答案  $g$  不正确，但你的操作方案合法且  $g \leq 10^4$ ，你同样可以获得 60% 的分数；你在每个子任务的得分是每个数据点得分的最小值。

1.  $n \leq 5$  (15 分)；



2.  $n \leq 10$  (15 分);
3. 第  $i(1 \leq i < n)$  条边连接  $V_i$  和  $V_{i+1}$  (15 分);
4. 第  $i(1 \leq i < n)$  条边连接  $V_1$  和  $V_{i+1}$  (15 分);
5. 第  $i(1 \leq i < n)$  条边连接  $V_{\lfloor \frac{i}{2} \rfloor}$  和  $V_{i+1}$  (10 分);
6. 每个点最多和 3 个其他顶点相连 (10 分);
7.  $n \leq 20$  (10 分);
8. 没有特殊性质 (10 分)。