

NOIP模拟赛

题目名称	保KDA	大相公	求排列	树形熔断器
题目类型	自定义比较器	全文比较	自定义比较器	全文比较
输入文件名	kda.in	dxg.in	roast.in	t.in
输出文件名	kda.out	dxg.out	roast.out	t.out
测试点数量	10	10	25	10
每个测试点时限	1s	4s	3s	4s
内存限制	256M	1024M	1024M	1024M

编译选项

对于 C++ 程序 `-lm -O2 -std=c++14`

A. 保KDA(kda)

题目描述

在《贪玩卓裕》这款游戏中，玩家们可以击杀别人获得击杀数，也可以协助队友击杀别人获得助攻数。当然，如果你被别人击杀，你也会获得死亡数。

衡量一个玩家战力的指标之一是 KDA(Kill Die Assist)，它等于 $\frac{\text{击杀数} + \text{助攻数}}{\max(\text{死亡数}, 1)}$ 。一个玩家的 KDA 越高，代表他的战斗力越高。

对于 nekko 这样精明的玩家来说，他知道一个亮眼的 KDA 可以带来别人的称赞。所以 nekko 的目标是最大化他的 KDA。

现在有 n 个不同的任务，进行第 i 个任务将会耗费 nekko t_i 分钟，而 nekko 的网费还剩下 m 分钟。在时间足够的情况下，nekko 可以自由选择进行哪些任务。

如果进行第 i 个任务，nekko 将会获得 a_i 击杀数、 b_i 死亡数、 c_i 助攻数。如果不进行第 i 个任务，nekko 会被队友惩罚，增加 d_i 的死亡数。

现在 nekko 想在下机前保一波 KDA。他想知道，他的 KDA 最大可能是多少。

输入格式

第一行两个整数 n, m 。

接下来 n 行，每行五个整数 t_i, a_i, b_i, c_i, d_i 。

输出格式

一行两个整数 p, q , 表示最大的 KDA 为 $\frac{p}{q}$ 。你需要保证 $\frac{p}{q}$ 为最简分数。若 KDA 最大为 0 , 则输出 0 1。

样例输入1

```
2 10
10 1 0 1 1
10 0 1 1 100
```

样例输出1

```
1 2
```

样例解释1

如果两个任务都不参加, $KDA = 0$ 。

如果只参加第一个任务, $KDA = \frac{1}{50}$ 。

如果只参加第二个任务, $KDA = \frac{1}{2}$ 。

由于时间不够, 无法同时参加两个任务。

样例输入2

```
20 100
49 1 1 6 2
17 7 1 4 0
17 8 2 3 0
13 4 0 9 2
21 3 1 2 0
1 7 0 3 0
30 4 0 5 1
21 5 1 4 2
22 8 1 4 2
11 5 0 7 0
8 4 2 5 0
22 3 0 9 2
4 4 0 5 1
27 9 0 9 1
49 6 0 5 0
9 2 1 5 0
16 9 1 3 1
1 1 1 2 0
11 8 2 5 2
21 1 0 2 1
```

样例输出2

```
87 10
```

样例输入3

```
1 100
100 1 0 1 0
```

样例输出3

```
2 1
```

样例解释3

为了避免除以 0，KDA 的计算方法为 $\frac{\text{击杀数} + \text{助攻数}}{\max(\text{死亡数}, 1)}$ 。

数据范围及限制

对于 30% 的数据，保证 $n \leq 20$ 。

对于额外 30% 的数据，保证对于每个任务， $b_i = 0$ ，并且 $d_i \times (a_i + c_i) = 0$ 。

对于额外 30% 的数据，保证存在一个任务，其 $b_i > 0, d_i > 0$ （也就是说 nekko 必死）。

对于所有 100% 的数据，保证

$1 \leq n, m \leq 1000, 0 \leq t_i \leq m, 0 \leq a_i, b_i, c_i, d_i \leq 1000, \sum a_i + c_i \leq 1000, \sum b_i + d_i \leq 1000$ 。

评分标准

各个测试点独立计算分数。在每个测试点中：

如果输出正确的 p, q ，可以获得全部分数。

你也可以不输出 p, q ，而只输出 -1 和 $\frac{p}{q}$ 的值（一个实数）。如果你输出的实数与正确答案的绝对误差或相对误差小于 10^{-5} ，则可以获得 70% 的分数。例如如果真实答案为 $\frac{1}{2}$ ，那么你可以输出 `-1 0.5`。只输出 `0.5` 会计 0 分。（输出 `-1` 是为了方便 spj）

B. 大相公(dxg)

题目描述

我们知道，麻将想要胡牌，手牌必须是四个面子和一个雀头（或七个对子、国士无双...但我们不管），共 14 张牌。其中面子是顺子 $(a, a+1, a+2)$ 或刻子 (a, a, a) 的三张牌，雀头是相同的两张牌。

打麻将时，大相公是指由于你出老干，手牌数大于 14。大相公原本是不能胡牌的（并且还要被剁手），但在本题中，只要一副手牌能被分为若干个面子和若干个雀头，就可以胡牌（例如 $\{1, 2, 3\}, \{2, 3, 4\}, \{2, 3, 4\}, \{5, 5, 5\}, \{6, 6, 6\}, \{7, 7\}, \{8, 8\}$ 可以胡牌）。

现在我们只考虑萬（万），不考虑条、筒、字牌，且每种牌不只有四张，而是无限张。你有 n 张手牌，有一部分给定，其余不知道是 1 ~ 9 萬中的哪一张。问你有多少种胡牌的手牌可能方式。我们只对集合计数，假设你的手牌是 $1, 2, ?, ?, ?$ ，那么 $1, 2, 3, 4, 4$ 和 $1, 2, 4, 4, 3$ 算一种。

输入格式

第一行一个整数 n ，表示手牌张数。

第二行一个长度为 n 的字符串表示你的手牌。

- 如果字符串中某个字符为 $1 \sim 9$ ，那么表示这张牌是对应的萬子。
- 如果字符串中某个字符为 $?$ ，那么表示这张牌是未知牌。

例如字符串 `12?3?1?22?` 表示你的手牌中有 6 张确定牌“一萬、一萬、二萬、二萬、二萬、三萬”，还有 4 张未知牌。而字符串 `???????` 则表示你的手牌中没有确定牌，只有 7 张未知牌。

输出格式

一行一个整数，表示答案对 998244353 取模后的值。

样例输入1

```
5
1234?
```

样例输出1

```
2
```

样例解释1

能够胡牌的手牌方案为 $1, 1, 2, 3, 4$ 或 $1, 2, 3, 4, 4$ 。其中 $1, 1, 2, 3, 4$ 可以被划分为 $1, 1$ 的雀头和 $2, 3, 4$ 的顺子； $1, 2, 3, 4, 4$ 可以被划分为 $4, 4$ 的雀头和 $1, 2, 3$ 的顺子。可以证明，其它手牌方案都不能胡牌。

样例输入2

```
3
???
```

样例输出2

```
16
```

样例解释2

能够胡牌的手牌方案为： $123/234/345/456/567/678/789/111/222/333/444/555/666/777/888/999$ ，共 16 种。可以证明，其它手牌方案都不能胡牌。

样例输入3

```
5
123??
```

样例输出3

10

样例解释3

两个 ? 除了可以组成 9 种雀头外，还能填入 1, 4（手牌为 1, 1, 2, 3, 4）。可以证明，其它手牌方案都不能胡牌。

样例输入4

14
1112345678999?

样例输出4

9

样例解释4

这是纯正九莲宝灯的牌型，? 可以是 1 ~ 9 万中的任意一张。

样例输入5

10
12257?????

样例输出5

78

样例输入6

50
????????????????????????????????????????????????

样例输出6

657355763

数据范围及限制

对于 30% 的数据，保证 $n \leq 5$ 。

对于额外 30% 的数据，保证字符串中 ? 的个数 ≤ 2 。

对于额外 30% 的数据，保证 $666 \leq n$ ，且字符串中每一位在 123456789? 这十种字符中等概率随机生成。

对于所有 100% 的数据，保证 $n \leq 10^5$ 。

C. 求排列(roast)

题目描述

给定 n 个数字 $a_1 \sim a_n$ 。

你需要求出一个排列 p ，使得在 $\max(a_{p_i} \oplus a_{p_{i+1}})$ 最小的情况下， p 的字典序最小。

输入格式

第一行一个整数 n 。

第二行 n 个整数 a_1, \dots, a_n 。

输出格式

一行 n 个整数，表示 p 。

样例输入1

```
8
1 9 2 6 0 8 1 7
```

样例输入1

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

样例输入2

```
13
1 1 4 5 1 4 1 9 1 9 8 1 0
```

样例输出2

```
1 2 3 4 5 6 7 8 9 10 11 13 12
```

样例输入3

```
30
0 0 1 2 0 2 1 2 1 2 1 0 1 1 2 1 0 2 2 1 1 1 2 0 2 2 1 1 2 2
```

样例输出3

```
1 2 3 5 4 6 8 10 12 7 9 11 13 14 16 17 15 18 19 23 25 26 29 30 24 20 21 22 27 28
```

数据范围及提示

对于 20% 的数据，保证 $n \leq 10$ 。

对于 50% 的数据，保证 $n \leq 1000$ 。

对于所有 100% 的数据, 保证 $n \leq 3 \times 10^5, 0 \leq a_i \leq 10^9$ 。

评分标准

各个测试点独立计算分数。在每个测试点中:

如果你输出了正确的 p , 则可以获得全部分数。

如果你没有输出正确的 p , 但是给出了使 $\max(a_{p_i} \oplus a_{p_{i+1}})$ 最小的一种可行解, 则可以获得本测试点 40% 的分数。

D. 树形熔断器(t)

题目描述

在人们的日常生活中, 电是不可或缺的一种能源。

但是由于焦耳定律 $Q = I^2 R t$, 当电流过大时, 很容易使产热过多, 引发火灾。

这时候我们需要一种电路元件——熔断器。熔断器是指当电流超过规定值时, 以本身产生的热量使熔体熔断, 断开电路的一种电器。常见的熔断器分为插入式熔断器、螺旋式熔断器、封闭式熔断器、快速熔断器、自复熔断器等等。

当熔断器和电容串联时, 就组成了“陈氏电流阻断器”, 类似于“外圈刮内圈烫”的思路, 当电压小时无法击穿电容, 电压大时又会使熔断器熔断, 所以可以阻挡任意大电压, 简直妙不可言。

现在 x 小区有 n 个住户, 编号分别为 $1 \sim n$ 。为了方便社区管理, 住户们形成了一个树形结构, 这样每个子树都可以形成一个子管理结构。

现在 x 小区通了电, 方便起见, 电线只在树上相邻节点间连通。安全起见, 每个住户都设置了一个熔断器, 第 i 号住户的熔断电压为 i 。

为了改善电路网的性能, 某个子管理结构 (也就是子树) 组织了一场测试, 测试具体如下:

- 子树中的每个住户都要抽签决定是否参加 (两个签, 分别为“参加”、“不参加”)。参加概率为 50%。
- 所有**参加**的住户都将接入电源, 并且**参加**的住户两两之间会形成电流, 也就是说**参加**的住户两两之间的路径上的所有住户都会通过电流。
- 测试开始时会从 0 开始逐渐增大电压, 当有住户的熔断器熔断时, 测试停止。本次测试的结果就是此时的电压。如果无人参加测试, 那么结果即为 0。

现在你需要回答测试结果的期望, 为了输出方便, 你只需要输出期望 $\times 2^S$ 对 998244353 取模后的值, 其中 S 表示子树的大小。

你还需要对每个子树都回答一次这个值。

输入格式

第一行一个正整数 n 。

第二行 $n - 1$ 个正整数, 第 i 个表示 $i + 1$ 号点的父亲 fa_{i+1} 。

不保证 $fa_i < i$ 。

输出格式

一行 n 个整数，第 i 个表示以 i 号点为根的子树的答案。

样例输入1

```
4
1 1 3
```

样例输出1

```
23 2 10 4
```

样例解释1

我们解释一下 10 这个数。3 号节点的子树中有两个点：3, 4。

如果无人参加测试，那么本次结果为 0。

如果只有 3 号节点参加，本次结果为 3。

如果只有 4 号节点参加，本次结果为 4。

如果 3, 4 号节点都参加，本次结果为 $\min(3, 4) = 3$ 。

那么 3 号节点子树的答案就是 $3 + 4 + 3 = 10$ 。

在整棵树中，如果 2, 3 号节点**参加**，那么 1, 2, 3 号节点都会通过电流，所以本次结果为 $\min(1, 2, 3) = 1$ 。

如果 2, 4 号节点**参加**，那么 1, 2, 3, 4 号节点都会通过电流，所以本次结果为 $\min(1, 2, 3, 4) = 1$ 。

样例输入2

```
7
1 2 1 4 1 4
```

样例输出2

```
161 7 3 32 5 6 7
```

样例输入3

```
7
5 5 1 1 5 1
```

样例输出3

```
165 2 3 4 44 6 7
```

样例输入4


```
40
1 2 2 1 5 5 6 7 8 5 8 6 12 14 2 12 4 9 2 17 10 19 16 10 18 7 7 13 28 6 23 31 21 13 15 5 27 4
32
```

样例输出4

```
746724168 1206 3 145 377625821 791331 7572 17023 492 97 11 1622 129 122 66 56 144 62 348 20
76 22 196 24 25 26 92 86 29 30 95 104 33 34 35 36 37 38 39 40
```

样例解释4

这是一个满足 $fa_i < i$ 的样例。

样例输入5

```
40
33 11 18 17 16 30 13 23 9 6 27 31 7 19 2 32 12 39 37 24 35 22 26 5 3 10 15 14 8 20 28 29 40
38 21 1 25 34 36
```

样例输出5

```
461424365 369145878 125876211 4 23027 427866113 503363763 134265224 457 249 226539515 58
973126285 897628308 299708 830519319 41471 26 594784 16826565 18889768 1745 887 37774699
13811 75544563 137 152172 595638376 713079049 654359719 78374 444643380 2365290 3474 9447304
738248980 6935 1184956 4725964
```

样例解释5

这是一个满足树是一条链的样例。

数据范围及限制

对于 10% 的数据，保证 $n \leq 10$ 。

对于 40% 的数据，保证 $n \leq 5000$ 。

对于额外 10% 的数据，保证 $fa_i < i$ 。

对于额外 30% 的数据，保证树是个链。

对于所有 100% 的数据，保证 $1 \leq n \leq 2 \times 10^5$ ， $1 \leq fa_i \leq n$ 。

提示：标算并不依赖于点 1 是根，这样只是为了读入方便。