

# NOI 2018 模拟题 D

## C\_SUNSHINE

题目名称	Dandelion	Dexterity	Dragonfly
源文件名	dandelion	dexterity	dragonfly
输入文件名	dandelion.in	dexterity.in	dragonfly.in
输出文件名	dandelion.out	dexterity.out	dragonfly.out
时间限制	1s	1s	2s
子任务数目	4	6	20
空间限制	666MB	666MB	666MB
题目类型	传统	传统	传统

### 注意事项：

1. 竞赛时间为 5 小时，比赛过程中不提供提交反馈，成绩以赛后统一评测为准。
2. 由于评测机配置可能不同，时间限制至少为标程最慢测试点运行时间的 1.5 倍上取整。
3. 最终评测时，打开 -O2 优化，C++ 语言使用 C++11 标准（编译参数加上 `--std=c++11`）。
4. C/C++ 语言下 64 位带符号整形请使用 `%lld` 输出。
5. 祝选手们比赛愉快。

## 1 Dandelion

### 1.1 问题描述

Lyra 做了个梦，梦见自己身处一片蒲公英花田里，拥有魔法的自己可以让蒲公英们可以实现自己的愿望。每个蒲公英有三个属性  $s_i, p_i, t_i$ ，对于第  $i$  株蒲公英，只有当 Lyra 的魔法值达到至少  $s_i$  的时候才有能力去完成这株蒲公英的愿望，当然需要花费她  $t_i$  天的时间，完成愿望后蒲公英作为奖励，在随风飘散的同时还会帮助 Lyra 继续提升自己，Lyra 的魔法值会增加  $p_i$ （注意帮蒲公英实现愿望不需要消耗魔法值）。

Lyra 初始时拥有魔法值  $R$ ，她想知道  $T$  天之内她最多能获得多少魔法值，你只需要输出她最终的能力值和完成任务的顺序即可。

### 1.2 输入格式

第一行三个整数  $n, T, R$ ，意义如题所述。

接下来  $n$  行，每行三个整数表示  $s_i, p_i, t_i$ 。

### 1.3 输出格式

第一行一个整数表示  $T$  天后 Lyra 的魔法值的最大值。

第二行若干个整数按次序给出 Lyra 依次完成愿望的蒲公英的编号。

### 1.4 样例输入输出

#### 1.4.1 样例输入 1

```
4 6 10
10 10 4
5 10 2
15 20 4
20 50 3
```

#### 1.4.2 样例输出 1

```
70
2 4
```

#### 1.4.3 样例输入 2

见 `/dandelion/ex_dandelion2.in`

**1.4.4 样例输出 2**

见 /dandelion/ex\_dandelion2.ans

**1.4.5 样例输入 3**

见 /dandelion/ex\_dandelion3.in

**1.4.6 样例输出 3**

见 /dandelion/ex\_dandelion3.ans

**1.5 数据范围与约定**

对于全部数据  $1 \leq n \leq 1000; 1 \leq t_i, T \leq 1000; 1 \leq s_i, R \leq 10^9; 1 \leq p_i \leq 10^6$ .

**Subtask 1[17pts]:**

$n \leq 9; t_i, T \leq 100; s_i, R \leq 10^9; p_i \leq 10^6$ .

**Subtask 2[23pts]:**

$n, T, R, s_i, p_i, t_i \leq 100$ .

**Subtask 3[21pts]:**

$t_i = 1$ .

**Subtask 4[39pts]:**

No Special Constraints.

## 2 Dexterity

Lyra 是一个灵巧的女孩子，她特别喜欢玩一种叫“石头剪刀布”的游戏，在这个游戏中，每回合双方同时打出一种手势，为石头 (r)，剪刀 (s)，布 (p) 之一，规定石头打败剪刀，剪刀打败布，布打败石头，若手势一样则视为平局。

虽然 Lyra 是一个灵巧的女孩子，她发现她依然赢不了 Evan，潜心研究多日 Evan 的策略后，发现在第二天的  $n$  轮游戏中，Evan 一定以某种固定策略出手势。

这个固定策略（一个长度为  $n$  的由 **r,s,p** 组成的字符串）就藏在 Evan 的电脑里，被 Evan 加密存储。Evan 的加密方式很奇怪，他先选取一个特定的  $d$ ，然后把整个字符串循环右移  $d$  个位置。

Lyra 拿到了加密后的策略串，她想在第二天的石头剪刀布比赛中大败 Evan，注意 Lyra 的策略不一定必须是开始前固定的，可以根据前若干回合的结果修正之后的策略。

在这场石头剪刀布大赛中，对于第  $i$  个回合，获胜可以获得  $w_i$  分，平局获得  $d_i$  分，而失败获得 0 分。Lyra 想知道自己采取最优策略的话，最坏情况下至少从这  $n$  个回合中获得多少分。

### 2.1 输入格式

第一行三个正整数  $n$  表示回合数。

第二行一个长度为  $n$  的字符串  $S$  表示 Evan 的序列的某个轮换。

接下来  $n$  行每行两个整数  $w_i, d_i$ ，表示获胜得分，平局得分。

### 2.2 输出格式

一行一个正整数表示 Lyra 至少获得的分数。

### 2.3 样例输入输出

#### 2.3.1 样例输入 1

```
5
rrsrr
3 1
3 1
3 1
3 1
3 1
```

#### 2.3.2 样例输出 1

```
12
```

### 2.3.3 样例输入 2

```
6
rsprsp
3 1
3 1
3 1
3 1
3 1
3 1
```

### 2.3.4 样例输出 2

```
15
```

### 2.3.5 样例输入 3

见 /dexterity/ex\_dexterity3.in

### 2.3.6 样例输出 3

见 /dexterity/ex\_dexterity3.ans

注意：对于所有的样例数据，获胜收益为 3，平局收益为 1。

## 2.4 数据范围与约定

对于全部数据  $1 \leq n \leq 1 \times 10^5; 0 \leq d_i \leq w_i \leq 10^9$ .

#### Subtask 1[5pts]:

$n \leq 10$ .

#### Subtask 2[14pts]:

$n \leq 16$ .

#### Subtask 3[17pts]:

$n \leq 50$ .

#### Subtask 4[18pts]:

$n \leq 1000$ .

**Subtask 5[19pts]:**

$n \leq 10^5$ ;  $S$  随机生成.

**Subtask 6[27pts]:**

$n \leq 10^5$ .

## 3 Dragonfly

### 3.1 问题描述

Lyra 新进入了一家自动飞行公司开始了实习, 作为一名同时精通算法, 系统, 硬件的天才, 她刚入职了两周就设计了一种自动飞行蜻蜓的模型。与无限撞玻璃的普通昆虫不同, 这种蜻蜓可以自动识别出房间的门并从中穿过。Lyra 甚至宣称, 这种蜻蜓还对路线有完整的把握, 可以时时刻刻知道自己在哪并规划路径。

为了演示这种蜻蜓的飞行技术, Lyra 设计了  $n$  个房间, 每个房间有左右两个门, 两个门走出去都会进入一个单向通道, 并从烟囱落入另一个房间, 换言之, 对于编号为  $i \in [1, n]$  的房间, 有两个属性  $L_i, R_i \in [1, n]$ , 若从左边的门出去则会飞到第  $L_i$  个房间, 而若从右边的门出去则会飞到第  $R_i$  个房间 (注意  $L, R$  不一定是排列)。

虽然房间系统非常复杂, Lyra 宣称她的电子蜻蜓依然可以在穿过错综复杂的通道系统后回到自己的房间, 可就在 Demo 前一天, Lyra 还没有开始写位置记录系统, 情急之下 Lyra 决定让蜻蜓按照如下的方法飞行:

重复  $A$  次: 走左边的门;  
再走右边的门;  
再重复  $B$  次: 走左边的门;  
再走右边的门;  
再重复  $C$  次: 走左边的门。

Lyra 发现只要自己精心设计房间的通道系统 (即  $L_i, R_i$ ), 电子蜻蜓无论从哪个房间出发, 经过这一系列操作后都会回到原来的房间, 这样就能混过 Demo 了。

Lyra 还很好奇一共有多少种不同的通道系统可以做到这样呢? 两种系统被认为不同当且仅当存在  $i$  使得对应的  $L_i$  或  $R_i$  不同。

### 3.2 输入格式

第一行一个整数  $T$  表示数据组数。

接下来  $T$  行每行四个整数  $n, A, B, C$ , 意义如题所述。

### 3.3 输出格式

对于每个输入输出一行一个整数表示答案对 998244353 取模的结果。

### 3.4 样例输入输出

#### 3.4.1 样例输入 1

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

20 13 18 100

### 3.4.2 样例输出 1

6

600

17287547

### 3.4.3 样例解释 1

对于第一组数据，可行的六种方案是：

- $L = \{1, 2, 3\}, R = \{1, 2, 3\}$
- $L = \{1, 2, 3\}, R = \{1, 3, 2\}$
- $L = \{1, 2, 3\}, R = \{2, 1, 3\}$
- $L = \{1, 2, 3\}, R = \{3, 2, 1\}$
- $L = \{2, 3, 1\}, R = \{1, 2, 3\}$
- $L = \{3, 1, 2\}, R = \{1, 2, 3\}$

## 3.5 数据范围及约定

对于 5% 的数据， $n \leq 5$ ;

对于 15% 的数据， $n \leq 8$ ;

对于 25% 的数据， $n \leq 10$ ;

对于 35% 的数据， $n \leq 20$ ;

对于 55% 的数据， $n \leq 50$ ;

对于 75% 的数据， $n \leq 400$ ;

对于 100% 的数据  $1 \leq T \leq 10; 1 \leq n \leq 1000; 0 \leq A, B, C \leq 10^{18}$ .