

简单模拟赛

Cards (cards, 2s, 512M, .in/.out)

大象正在玩一个叫塔尖戮杀的游戏。他目前有 n 张手牌，第 i 张牌可以选择两种效果之一：加 a_i 力量或完成一次伤害 b_i 的攻击。大象一开始的力量是 0。如果你打出的手牌原始攻击为 a 且当前有 x 的力量，实际攻击伤害就会是 $a + x$ 。他总共打出的伤害是所有手牌卡牌造成的伤害之和。

他已经计划好了按什么顺序打这些牌，但是还没有想好每张牌触发哪个效果，于是他找到了你。请输出最优可以造成的伤害和任意一种方案。

输入格式

第一行一个正整数 n ，表示手牌的数量。

接下来的 n 行，每行包含两个非负整数 a_i 及 b_i ，描述每张牌的两种效果。

输出格式

第一行输出一个整数，表示最大可能的伤害和。

第二行输出 n 个整数，表示对应手牌使用的效果，1 表示选择加力量效果，2 表示选择攻击效果。

如果有多种伤害最大的方案，输出任意一种。

样例输入1

```
4
5 1
4 1
2 3
15 1
```

样例输出1

```
22
1 1 2 2
```

一种最优策略为，对于第一张卡触发加力量效果，力量提升为 5；对于第二张卡触发加力量效果，力量提升为 9；对于第三张卡触发攻击，实际伤害为 $9 + 3 = 12$ ；对于第四张卡触发攻击，实际伤害为 $9 + 1 = 10$ 。总伤害 22。

样例输入2

```
3
1 10
10 100
100 1
```

样例输出2

```
111
2 2 2
```

一种最优策略为三张牌均触发攻击效果。

样例输入3、样例输出3

见选手文件夹下 `ex_cards3.in` 和 `ex_cards3.ans`。50。爆int。

样例输入4、样例输出4

见选手文件夹下 `ex_cards4.in` 和 `ex_cards4.ans`。5000。爆int。

数据范围

对于所有数据， $1 \leq n \leq 5000$ ， $0 \leq a_i, b_i \leq 10^9$ 。

对于 10% 的数据， $n \leq 20$ 。

对于 30% 的数据， $n \leq 50$ ， $a_i, b_i \leq 50$ 。

对于 50% 的数据， $n \leq 50$ 。

对于 70% 的数据， $n \leq 400$ 。

Meet (meet, 3s, 512M, .in/.out)

长颈鹿国的每个市选出了一位市议员，他们即将发表就职演讲。

长颈鹿国的结构可以由一棵树描述： n 个城市，有 $n - 1$ 条双向道路连接所有城市，第 i 条道路连接城市 a_i 和 b_i ，使用这条道路通勤需要时间 t_i 。长颈鹿市计划在两个城市设立会场，每个市议员可以任选一个会场参与，并从自己的城市使用道路到达那个会场。大象希望你来选择这两个城市，使得最优情况下每个市议员的通勤时间总和最小（只考虑单程的时间）。

输入格式

第一行包含一个正整数，表示城市的数量 n 。

接下来 $n - 1$ 行，每行三个正整数 a_i 、 b_i 和 t_i ，表示道路连接的两个城市和使用这条道路需要的时间。

输出格式

输出一个正整数，表示如果选取了最佳的两个会场位置，市议员通勤时间的最小总和。

样例输入1

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

样例输出1

```
12
```

一种最优策略为选择城市 2、3 作为会场，每个城市的市议员分别需要 $\{3, 0, 0, 5, 4\}$ 的通勤时间。

样例输入2、样例输出2

见选手文件夹下 `ex_meet2.in` 和 `ex_meet2.ans`。100。

样例输入3、样例输出3

见选手文件夹下 `ex_meet3.in` 和 `ex_meet3.ans`。200000。

数据范围

对于所有数据， $2 \leq n \leq 100000$ ， $1 \leq t_i \leq 10^6$ 。

对于 20% 的数据， $n \leq 100$ 。

对于 50% 的数据， $n \leq 2000$ 。

对于另外 20% 的数据，保证树是一条 $1 \rightarrow 2 \rightarrow \dots \rightarrow n$ 的链，亦即输入中对于每个 $i \in [1, n - 1]$ 都存在一条 i 连向 $i + 1$ 的边。

Subseq (subseq, 2s, 512M, .in/.out)

大象很喜欢子序列。他有一个由小写字母组成的字符串，他想知道这个字符串中有多少个 **只出现一次的** 子序列，对 998244353 取模。

输入格式

输入有多组数据，第一行一个正整数 T 表示数据组数。

对于每组数据，输入一行一个由小写字母组成的字符串 s 。

输出格式

对于每组数据输出一个 $[0, 998244352]$ 的整数，表示字符串 s 中有多少个 **只出现一次的** 子序列，对 998244353 取模。

样例输入1

```
3
aabb
subsequence
ishard
```

样例输出1

```
3
1343
63
```

在第一组数据中只出现一次的子序列有 $\{aa, bb, aabb\}$ 。有其他子序列，但是都出现了至少两次。

样例输入2、样例输出2

见选手文件夹下 `ex_subseq2.in` 和 `ex_subseq2.ans`。10。

样例输入3、样例输出3

见选手文件夹下 `ex_subseq3.in` 和 `ex_subseq3.ans`。500。

数据范围

对于所有数据, $T \leq 10$, $1 \leq |s| \leq 200000$ 。

对于 10% 的数据, $1 \leq |s| \leq 15$ 。

对于 30% 的数据, $1 \leq |s| \leq 100$ 。

对于 50% 的数据, $1 \leq |s| \leq 2000$ 。

对于 70% 的数据, $1 \leq |s| \leq 20000$ 。

Sort (sort, 3s, 512M, .in/.out)

大象有一个排序算法：

```
def incorrect_sort(x):
    for i=1,2...|x|:
        m=i                # (*)
        for j=1,2...|x|:
            if x[m]>x[j]:
                m=j          # (*)
        swap(x[i],x[m])
```

这个算法不一定对，但是不重要。你需要计算对于给定 1 到 n 的排列 a_1, a_2, \dots, a_n ，**每个前缀** a_1, a_2, \dots, a_i 调用 `incorrect_sort` 时 m 会被赋值几次（即被标注星号的行的被执行次数）。

输入格式

第一行一个正整数 n ，表示序列长度。

第二行 n 个正整数 a_1, a_2, \dots, a_n ，表示序列元素。

输出格式

输出一行 n 个非负整数，表示对 **每个前缀** 调用 `incorrect_sort` 时 m 被赋值的次数。

样例输入1

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

样例输出1

```
1 3 6 8 13 17
```

样例输入2

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

样例输出2

1 4 8 11 13 19

样例输入3、样例输出3

见选手文件夹下 `ex_sort3.in` 和 `ex_sort3.ans`。500。

样例输入4、样例输出4

见选手文件夹下 `ex_sort4.in` 和 `ex_sort4.ans`。500000。

数据范围

对于所有数据, $1 \leq n \leq 500000$, $1 \leq a_i \leq n$, 所有 a_i 两两不同。

对于 10% 的数据, $n \leq 500$ 。

对于 40% 的数据, $n \leq 4000$ 。

对于另外 30% 的数据, 保证 a_1, a_2, \dots, a_n 在所有可能排列中等概率均匀随机。