

PION 2024 模拟赛

HEZ

时间：2024 年 10 月 15 日 08:30 ~ 13:00

题目名称	冒泡	山路	机器	旅行
题目类型	传统型	传统型	传统型(spj)	传统型
目录	pop	road	machine	tour
可执行文件名	pop	road	machine	tour
输入文件名	pop.in	road.in	machine.in	tour.in
输出文件名	pop.out	road.out	machine.out	tour.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
子任务数目	5	6	6	8
测试点是否等分	否	否	否	否

提交源程序文件名

对于 C++ 语言	pop.cpp	road.cpp	machine.cpp	tour.cpp
-----------	---------	----------	-------------	----------

编译选项

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

注意事项（请仔细阅读）

- 文件名（程序名和输入输出文件名）必须使用英文小写，**不需要建立子文件夹**。
- 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。
- 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
- 选手提交的程序源文件必须不大于 100KB。
- 程序可使用的栈空间内存限制与题目的内存限制一致。
- 只提供 Linux 格式附加样例文件**。
- 对于输入子任务编号的题目，下发样例文件中的编号表示**其满足哪个子任务的限制**。
- 禁止在源代码中改变编译器参数（如使用 `#pragma` 命令），禁止使用系统结构相关指令（如内联汇编）和其他可能造成不公平的方法。
- 评测时采用的机器配置为：Windows 11, i7-12700KF @3.60GHz, 32G, Lemonlime。使用的编译器版本为：**x86_64-pc-msys GCC 13.3.0**。上述时限以此配置为准。

冒泡 (pop)

【题目背景】

小 M 刚学会冒泡排序。

【题目描述】

给定两个长度为 N 的排列 A, B ，小 M 需要重新排列 A 中的元素，使得 $\text{value} = \sum_{i=1}^n |A_i - B_i|$ 最大。

但是由于小 M 只会冒泡，所以他每次操作只能交换 A 中 **相邻** 的两个元素。

你需要在最大化 value 的前提下，最小化操作步数。

你只需要输出最小的操作步数即可。

【输入格式】

从文件 *pop.in* 中读入数据。

输入的第一行包含一个正整数 N 。

第二行，输入 N 个数，表示 A_i 。

第三行，输入 N 个数，表示 B_i 。

【输出格式】

输出到文件 *pop.out* 中。

输出一行一个整数，表示最小步数。

【样例 1 输入】

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

【样例 1 输出】

```
1 1
```

【样例 2 输入】

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

【样例 2 输出】

```
1 3
```

【样例 3】

见选手目录下的 *pop/pop3.in* 与 *pop/pop3.ans*。
这个样例满足子任务 2 的约束条件。

【样例 4】

见选手目录下的 *pop/pop4.in* 与 *pop/pop4.ans*。
这个样例满足子任务 3 的约束条件。

【样例 5】

见选手目录下的 *pop/pop5.in* 与 *pop/pop5.ans*。
这个样例满足子任务 4 的约束条件。

【样例 6】

见选手目录下的 *pop/pop6.in* 与 *pop/pop6.ans*。
这个样例满足子任务 5 的约束条件。

【数据范围】

对于所有测试数据保证： $1 \leq N \leq 10^6$ ， A, B 均为 $1 \sim N$ 的排列。

子任务编号	$N \leq$	特殊性质	分值
1	8	无	5
2	20		10
3	5000		20
4	10^6	N 为偶数	15
5	10^6	无	50

山路 (road)

【题目描述】

杭州西湖边的群山上 有 N 个景点，景点之间通过若干条 无向 的道路连接，每条道路有一个正整数长度。也就是说，我们可以将其看作一张 无向简单图。

小 μ 希望从某个景点出发，经过 至少一条 道路，最后回到这个景点，使得经过的道路总长最短。此外，他还要求经过的 道路 没有重复。

你需要对每个景点都求出答案。也就是说，对于每个顶点，你需要求出包含它的 最小简单环 长度。

【输入格式】

从文件 `road.in` 中读入数据。

第一行，输入两个正整数 `sub, N`，其中 `sub` 表示该组数据所属的子任务编号。

接下来输入 $N - 1$ 行，第 i 行共有 $N - i$ 个数 $G_{i,j}$ ，描述这张图。其中，如果景点 (i,j) 之间没有道路，那么 $G_{i,j} = -1$ ；否则 $G_{i,j}$ 表示连接景点 (i,j) 的道路的长度。

【输出格式】

输出到文件 `road.out` 中。

输出一行 N 个数，第 i 个数表示景点 i 的答案。

特别地，如果无解，在对应位置输出 `-1`。

【样例 1 输入】

```
1 2 4
2 9 1 1
3 -1 1
4 -1
```

【样例 1 输出】

```
1 11 11 -1 11
```

【样例 2】

见选手目录下的 `road/road2.in` 与 `road/road2.ans`。
这个样例满足子任务 1 的约束条件。

【样例 3】

见选手目录下的 `road/road3.in` 与 `road/road3.ans`。
这个样例满足子任务 2 的约束条件。

【样例 4】

见选手目录下的 `road/road4.in` 与 `road/road4.ans`。
这个样例满足子任务 3 的约束条件。

【样例 5】

见选手目录下的 `road/road5.in` 与 `road/road5.ans`。
这个样例满足子任务 4 的约束条件。

【样例 6】

见选手目录下的 `road/road6.in` 与 `road/road6.ans`。
这个样例满足子任务 5 的约束条件。

【样例 7】

见选手目录下的 `road/road7.in` 与 `road/road7.ans`。
这个样例满足子任务 6 的约束条件。

【数据范围】

对于所有测试数据保证： $1 \leq N \leq 300, G_{i,j} \in [-1, -1] \cup [1, 10^6]$ 。

子任务编号	$N \leq$	特殊性质	分值
1	300	A	10
2	10	无	10
3	50		10
4	100		20
5	300	B	20
6		无	30

特殊性质 A：保证给出的图是基环森林。

特殊性质 B：输入数据随机生成。随机方式为，给定 N ， $G_{i,j}$ 有 25% 的概率为 -1 ，75% 的概率在 $[1, 10^6]$ 中均匀随机选取。

机器 (machine)

【题目背景】

小 M 正在研究一种神秘的机器。

【题目描述】

小 M 面前有一排弹珠，弹珠有红色(R)和蓝色(B)两种颜色，分别重 1 克，2 克。小 M 的机器 **每次** 可以选择一个长度 **不超过 3** 的区间，然后翻转这个区间内的弹珠，花费的代价为 **区间内弹珠重量之和** 加上 C ，其中 C 是一个给定的常数。

例如，对区间 $[R,R,B]$ 操作后，区间变为 $[B,R,R]$ ，并且花费 $4 + C$ 的代价；对区间 $[B,R]$ 操作后，区间变为 $[R,B]$ ，并且花费 $3 + C$ 的代价。

你要求出一种使用机器的方案，将这排弹珠重排成给定状态，使得 **总代价** 最小。数据保证一定有解。

【输入格式】

从文件 *machine.in* 中读入数据。

第一行输入三个整数 sub, N, C ，其中 sub 表示该组数据所属的子任务编号。

第二行，输入一个长度为 N 的字符串 S ，每个字符为 $\{R, B\}$ 之一，表示初始的弹珠序列。

第三行，输入一个长度为 N 的字符串 T ，每个字符为 $\{R, B\}$ 之一，表示初始的弹珠序列。

保证 S 和 T 中 R, B 的数量分别相同。

【输出格式】

输出到文件 *machine.out* 中。

在第一行，输出一个整数 m ，表示你构造的方案步数。你需要保证 $0 \leq m \leq 10^6$ ，可以证明，一定存在一种总代价最小的方案满足这一要求。

接下来 m 行，每行两个整数 l, r ，表示每次操作的区间。你需要保证 $1 \leq l \leq r \leq n$ ，并且 $r \leq l + 2$ 。

如果有多种方案，你可以输出任意一种。

注意：你不需要最小化 m 。你只需要最小化总代价。

【 样例 1 输入 】

```
1 1 5 2
2 RRBBR
3 BRRRB
```

【 样例 1 输出 】

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

【 样例 1 解释 】

操作如下：

- 1. 翻转 [1,3]，序列变为 **BRRBR**，代价为 $2 + 1 + 1 + 2 = 6$ 。
- 2. 翻转 [4,5]，序列变为 **BRRRB**，代价为 $2 + 1 + 2 = 5$ 。

总代价为 11，可以证明不存在一组代价更小的方案。

【 样例 2 输入 】

```
1 1 7 0
2 BBRBRBR
3 RBRBRBB
```

【 样例 2 输出 】

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

【 样例 2 解释 】

最小代价为 14。

【样例 3】

见选手目录下的 *machine/machine3.in* 与 *machine/machine3.ans*。
这个样例满足子任务 2 的约束条件。

【样例 4】

见选手目录下的 *machine/machine4.in* 与 *machine/machine4.ans*。
这个样例满足子任务 3 的约束条件。

【样例 5】

见选手目录下的 *machine/machine5.in* 与 *machine/machine5.ans*。
这个样例满足子任务 4 的约束条件。

【样例 6】

见选手目录下的 *machine/machine6.in* 与 *machine/machine6.ans*。
这个样例满足子任务 5 的约束条件。

【样例 7】

见选手目录下的 *machine/machine7.in* 与 *machine/machine7.ans*。
这个样例满足子任务 6 的约束条件。

【数据范围】

对于所有数据，保证 $1 \leq N \leq 1000, 0 \leq C \leq 1000$ ， S 和 T 仅包含 R, B 两种字符，且 R, B 的数量分别相同。

子任务编号	$N \leq$	特殊性质	分值
1	20	无	10
2	50	A	20
3	300	无	20
4	500	B	15
5		C	15
6	1000	无	20

特殊性质 A：保证 S 中蓝色弹珠的数量不超过 20。
特殊性质 B：保证 S 和 T 中所有蓝色弹珠的下标都具有相同的奇偶性。
特殊性质 C：保证 S, T 中有至少一个蓝色弹珠。设 u_i, v_i 分别为 S, T 中第 i 个蓝色弹珠，有 $\max_i\{u_i\} < \min_i\{v_i\}$ 。

【提示】

本题下发校验器。对于样例 3 ~ 8，答案文件中没有给出具体的方案。

具体来说，这些样例的第一行均为 -1，第二行为最小代价。

你可以将下发文件中 *machine/checker.cpp* 和 *machine/testlib.h* 复制到同一目录下，运行 `g++ checker.cpp -o checker -std=c++14 -O2` 编译本题的校验器。

你可以根据你使用的系统，选择以下命令之一运行校验器：

- 对于类 UNIX 系统（如：Linux，macOS）

```
1 ./checker <input> <output> <answer>
```

- 对于 Windows 系统：

```
1 .\checker.exe <input> <output> <answer>
```

其中 `<input>` `<output>` `<answer>` 分别是你需要测试的输入文件、选手输出文件和答案文件。

注意：下发校验器可能和评测时的校验器不同。选手需要保证输入文件和答案文件合法，否则不保证校验器可以正常运行。

旅行 (tour)

【题目背景】

小 μ 正在外地旅行。由于他已经厌倦了找环，他正在一个树形景区里游走。

【题目描述】

给定一棵 N 个点，以 1 为根的有根树。每个点有一个权值 a_i ，表示小 μ 对它的喜爱程度。

小 μ 的旅行共包含 T 天。每一天的日程是以下四种之一：

1. **回忆**。给定一个点 x ，小 μ 想要知道他对这个点的喜爱程度。由于答案可能很大，请输出其对 $P = 998244353$ 取模后的结果。
2. **跋涉**。给定两个点 x, y 和参数 k, b ，对于每个在 x 到 y 路径上的点 p ，执行 $a_p \leftarrow ka_p + b$ 。
3. **打卡**。给定一个点 x 和参数 k, b ，对于每个 x 子树内的点 p ，执行 $a_p \leftarrow ka_p + b$ 。
4. **游走**。给定一个点 x 和参数 r, k, b ，对于每个距离 x 不超过 r 的点 p ，执行 $a_p \leftarrow ka_p + b$ 。特别地，由于小 μ 不想迷路，所以 r 不会太大，详见数据范围。本题中，树上两点 x, y 之间的距离定义为 x 到 y 的最短路径的边数。

【输入格式】

第一行输入三个正整数 sub, N, T ，其中 sub 表示该组数据所属的子任务编号。

接下来 $N - 1$ 行，每行两个正整数 u, v ，描述一条树边。

接下来一行 N 个整数 a_i ，表示每个点的初始喜爱程度。

接下来 T 行，每行以一个整数 o_i 开头，表示这一天的日程：

- 如果 $o_i = 1$ ，则输入一个整数 x_i 。
- 如果 $o_i = 2$ ，则输入四个整数 x_i, y_i, k_i, b_i 。
- 如果 $o_i = 3$ ，则输入三个整数 x_i, k_i, b_i 。
- 如果 $o_i = 4$ ，则输入四个整数 x_i, r_i, k_i, b_i 。
- 上述变量含义如题面所述。

【输入格式】

对于所有 $o_i = 1$ 的日子，输出一行，表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 47
2 8
```

【样例 2】

见选手目录下的 *tour/tour2.in* 与 *tour/tour2.ans*。
这个样例满足子任务 1 的约束条件。

【样例 3】

见选手目录下的 *tour/tour3.in* 与 *tour/tour3.ans*。
这个样例满足子任务 1 的约束条件。

【样例 4】

见选手目录下的 *tour/tour4.in* 与 *tour/tour4.ans*。
这个样例满足子任务 2 的约束条件。

【样例 5】

见选手目录下的 *tour/tour5.in* 与 *tour/tour5.ans*。
这个样例满足子任务 3 的约束条件。

【样例 6】

见选手目录下的 *tour/tour6.in* 与 *tour/tour6.ans*。
这个样例满足子任务 4 的约束条件。

【样例 7】

见选手目录下的 `tour/tour7.in` 与 `tour/tour7.ans`。
这个样例满足子任务 5 的约束条件。

【样例 8】

见选手目录下的 `tour/tour8.in` 与 `tour/tour8.ans`。
这个样例满足子任务 6 的约束条件。

【样例 9】

见选手目录下的 `tour/tour9.in` 与 `tour/tour9.ans`。
这个样例满足子任务 7 的约束条件。

【样例 10】

见选手目录下的 `tour/tour10.in` 与 `tour/tour10.ans`。
这个样例满足子任务 8 的约束条件。

【数据范围】

对于所有数据，保证 $1 \leq N, T \leq 10^5$ ， $1 \leq x_i, y_i \leq N$ ， $0 \leq r_i \leq 10$ ， $o_i \in \{1, 2, 3, 4\}$ ， $0 \leq a_i, k_i, b_i < P$ 。

子任务编号	N, T	r_i	o_i	k_i	分值
1	≤ 1000	≤ 10	$\in \{1, 2, 3, 4\}$	$\in [0, P)$	10
2	$\leq 10^5$		$\in \{1, 4\}$		10
3		$= 1$	$\in \{1, 3, 4\}$		15
4		无	$\in \{1, 2, 3\}$		15
5		$= 1$	$\in \{1, 2, 3, 4\}$	15	
6		≤ 10		$= 1$	15
7	$\leq 10^4$			$\in [0, P)$	10
8	$\leq 10^5$				10

【提示】

请选手相信自己代码的常数。