

NOIP2024 模拟赛

GDFZ

时间：2024 年 11 月 25 日

题目名称	树	序列	转圈	游戏
题目类型	传统题	传统题	传统题	传统题
目录	tree	seq	circ	game
可执行文件名	tree	seq	circ	game
输入文件名	tree.in	seq.in	circ.in	game.in
输出文件名	tree.out	seq.out	circ.out	game.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
内存限制	512 MB	1024 MB	512 MB	1024 MB
子任务数目	10	11	20	9
测试点是否等分	是	否	是	否

提交源程序文件名

对于 C++ 语言	tree.cpp	seq.cpp	circ.cpp	game.cpp
-----------	----------	---------	----------	----------

编译选项

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

注意事项（请仔细阅读）

- 1. 测试机器：CPU(AMD Ryzen 5 3600 6-Core Processor *12), RAM 8.0G。
- 2. 系统环境：NOI Linux 2.0(基于 Ubuntu 20.04.1)。
- 3. 文件名（程序名和输入输出文件名）必须使用英文小写。
- 4. C/C++ 中函数 `main()` 返回类型必须是 `int`，程序正常结束返回值必须是 0。
- 5. 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）。
- 6. 选手提交的程序源文件必须不大于 100KB。
- 7. 程序可使用的栈空间内存限制与题目的内存限制一致。

树 (tree)

【题目描述】

对于一棵树，记 $f(i) = \sum_{1 \leq j \leq n} dis(i, j)$ ，其中 $dis(i, j)$ 表示在树上 i, j 之间的距离。

给定一个数 x ，你需要构造一棵树，满足存在两个点 u, v 满足 $f(u) - f(v) = x$ ，同时树的节点数最少。

方便起见，你只需要输出这个最小值即可。

【输入格式】

从文件 `tree.in` 中读入数据。

本题单个测试点内有多组测试数据，第一行一个整数 T ，表示数据组数。

对于每组数据，一个整数 x 。

【输出格式】

输出到文件 `tree.out` 中。

对于每组数据，输出所需的最小节点数。

【样例 1 输入】

```
1 3
2 2
3 3
4 114514
```

【样例 1 输出】

```
1 4
2 5
3 678
```

【样例 2】

见选手目录下的 `tree/tree2.in` 与 `tree/tree2.ans`。

满足测试点 3 ~ 4 的限制。

【样例 4】

见选手目录下的 *tree/tree4.in* 与 *tree/tree4.ans*。
分别满足测试点 7 ~ 8 的限制。

【数据范围】

对于全部数据，满足 $1 \leq T \leq 10^5, 0 \leq x \leq 10^{18}$ 。

子任务编号	特殊性质	子任务分值
1	$x \leq 10$	20
2	$x \leq 10^3$	20
3	$x \leq 10^6$	20
4	$x \leq 10^9$	20
5	无	20

序列 (seq)

【题目描述】

小 Q 很喜欢基因序列问题, 按照剧本, 你有两个长度分别为 n 和 m 的 0/1 串 S 与 T 。你需要在 S 选择恰好 m 个位置 p_1, p_2, \dots, p_m , 满足:

- $1 \leq p_1 < p_2 < \dots < p_m \leq n$, 即选择的 m 个位置必须单调递增。
- $T = S_{p_1} S_{p_2} \dots S_{p_m}$, 即这 m 个位置按顺序写下可得到串 T 。
- 对任意 $0 \leq i \leq n$, 记 $l = p_i + 1, r = p_{i+1} - 1$, 则 $S[l \dots r]$ 仅由一种字符组成。
 - 额外规定 $p_0 = 0$ 且 $p_{n+1} = n + 1$ 。
 - 注意当 $l > r$ 时, $S[l \dots r]$ 为空串, 仍然符合要求。

虽然这和基因序列问题没什么关系, 但是小 Q 想要知道, 是否存在任意一组合法方案, 能够选出这 m 个位置。如果存在, 你需要给出任意一种合法的方案。如果不存在, 你需要报告不存在任何合法的方案。

【输入格式】

从文件 `seq.in` 中读入数据。
输入的第一行包含一个串 S , 仅由数字 0/1 构成。
输入的第二行包含一个串 T , 仅由数字 0/1 构成。

【输出格式】

输出到文件 `seq.out` 中。
若不存在合法的方案, 则输出一行一个字符串 No。
否则, 输出的第一行包含一个字符串 Yes。
接下来一行, 包含 m 个整数 p_1, p_2, \dots, p_m , 描述任意一组合法的方案。

【样例 1 输入】

```
1 0001101101
2 0110
```

【样例 1 输出】

```
1 YES
2 3 5 7 9
```

【样例 1 解释】

样例提供了一组合法的方案，满足 $p_1 = 3, p_2 = 5, p_3 = 7, p_4 = 9$ 。容易验证 $S_3S_5S_7S_9 = 0110 = T$ ，满足第一个条件。

对于第二个条件：

- 当 $i = 0$ 时, $S[1 \cdots 2] = 00$, 符合条件。
- 当 $i = 1$ 时, $S[4 \cdots 4] = 1$, 符合条件。
- 当 $i = 2$ 时, $S[6 \cdots 6] = 0$, 符合条件。
- 当 $i = 3$ 时, $S[8 \cdots 8] = 1$, 符合条件。
- 当 $i = 4$ 时, $S[10 \cdots 10] = 1$, 符合条件。

【样例 2 输入】

```
1 00
2 1
```

【样例 2 输出】

```
1 NO
```

【样例 3 输入】

```
1 11101110111011101110111
2 00000
```

【样例 3 输出】

```
1 YES
2 4 8 12 16 20
```

【样例 4】

见选手目录下的 `seq/seq4.in` 与 `seq/seq4.ans`。

【数据范围】

对于 100% 的数据, 保证 $1 \leq m \leq n \leq 10^5$, S, T 只由字符 0 与 1 组成。

子任务编号	$n \leq$	m	特殊性质	分值
1	≤ 20	$\leq n$	无	3
2	≤ 30			6
3	≤ 40		A	7
4	≤ 50			6
5	≤ 300			12
6	≤ 2000		B	10
7		≤ 10	无	7
8		$\leq n$		13
9			B	12
10	$\leq 10^5$	≤ 20	无	9
11		$\leq n$		15

- 性质 A: 保证串 T 随机生成。
- 性质 B: 保证对任意 $1 \leq i < m$, 都有 $T_i \neq T_{i+1}$ 。

转圈 (circ)

【题目描述】

给定一个 n 个点 m 条边的有向图（可以有重边自环），你可以进行如下操作：

- 选定两个不同的点 x, y ;
- 将 x 合并到 y 上，即将所有边 $z \rightarrow x$ 改为 $z \rightarrow y$ ，所有 $z \leftarrow x$ 改为 $z \leftarrow y$ 。

求至少进行多少次操作能使得图变为一个环。

定义一个 n 个点有向图是环当且仅当，存在一种标号方式，使得， $E = \{i \rightarrow (i + 1) \bmod n \mid 0 \leq i < n\}$ 。

【输入格式】

从文件 `circ.in` 中读入数据。

本题有多组测试数据。

输入第一行有一个整数 T ，表示数组组数。

对于每一组数据，输入共 $m + 1$ 行，第 1 行包含两个整数 n, m ，表示图的点数和边数。
接下来 m 行，每行两个整数 u, v ，表示图中的一条边。

【输出格式】

输出到文件 `circ.out` 中。

对于每一组数据，输出一个整数，表示最少操作次数，让原图变成一个环。

【样例 1 输入】

```
1 4
2 3 4
3 0 1
4 1 2
5 2 0
6 0 1
7 3 5
8 0 0
9 1 2
10 2 0
11 0 1
```

```
12 0 1
13 4 2
14 0 1
15 2 3
16 6 6
17 0 2
18 1 5
19 3 0
20 4 0
21 1 4
22 5 0
```

【样例 1 输出】

```
1 0
2 2
3 2
4 3
```

【样例 1 解释】

对于第一组数据, 它已经是一个环了, 所以不需要操作;
对于第二组数据, 只能将它缩成单个点;
对于第三组数据, 可以对点对 $(1, 2)$ 和点对 $(3, 0)$ 各操作一次变成一个大小为 2 的环。

【样例 2】

见选手目录下的 *circ/circ2.in* 与 *circ/circ2.ans*。

【样例 3】

见选手目录下的 *circ/circ3.in* 与 *circ/circ3.ans*。

【数据范围】

对于所有数据 $T \leq 5, 1 \leq n \leq 10^5, 1 \leq m \leq 2 \times 10^5$ 。

子任务编号	$n \leq$	$m \leq$	特殊性质	子任务分值
1	5	20	无	20
2	10^3	3×10^3	无	30
3	10^5	2×10^5	A	10
4	10^5	2×10^5	无	40

特殊性质 A: $m = n - 1$ ，且图是一个弱连通块。
由于输入量很大, 请考虑使用高效的输入方式。

游戏 (game)

【题目描述】

给定一个由非负整数构成的序列 a_1, a_2, \dots, a_n , 要求通过最少的以下操作, 使得 $\forall 1 \leq i \leq n, a_i = b_i$ 。

- 选定 l, r , 对所有 $[l, r]$ 中满足 $a_i > 0$ 的 i , 令 $a_i \leftarrow a_i + 1$.
- 选定 l, r , 对所有 $[l, r]$ 中满足 $a_i > 0$ 的 i , 令 $a_i \leftarrow a_i - 1$.

【输入格式】

从文件 *game.in* 中读入数据。
输入中包含多组测试数据。
输入的第一行包含一个整数 T , 表示数据组数。
对于每组数据, 第一行一个整数 n .
接下来一行, 包含 n 个整数, 描述 a_1, a_2, \dots, a_n .
接下来一行, 包含 n 个整数, 描述 b_1, b_2, \dots, b_n .

【输出格式】

输出到文件 *game.out* 中。
对于每组数据, 输出一行一个整数, 表示答案. 若合法方案不存在, 则输出 -1.

【样例 1 输入】

```
1 3
2 5
3 1 1 1 1 1
4 2 0 2 0 2
5 6
6 1 1 4 5 1 4
7 1 9 1 9 8 10
8 8
9 2 0 0 4 1 1 1 5
10 2 0 0 5 0 1 1 7
```

【样例 1 输出】

```
1 3
2 18
3 4
```

【样例 1 解释】

对于第一组数据, 最优的策略为:

- 选择区间 $[2, 2]$ 将其减 1 , 序列变为 $[1, 0, 1, 1, 1]$
- 选择区间 $[4, 4]$ 将其减 1 , 序列变为 $[1, 0, 1, 0, 1]$
- 选择区间 $[1, 5]$ 将其加 1 , 序列变为 $[2, 0, 2, 0, 2]$

【样例 2】

见选手目录下的 *game/game2.in* 与 *game/game2.ans*。
该样例满足子任务 1 的特殊性质。

【样例 3】

见选手目录下的 *game/game3.in* 与 *game/game3.ans*。
该样例满足子任务 2 的特殊性质。

【样例 4】

见选手目录下的 *game/game4.in* 与 *game/game4.ans*。
该样例满足子任务 9 的特殊性质。

【数据范围】

本题采用捆绑测试。
对于所有数据, $1 \leq n \leq 10^6, 1 \leq \sum n \leq 3 \times 10^6, 0 \leq a_i, b_i \leq 10^9$ 。

子任务编号	$n \leq$	$a_i, b_i \leq$	特殊性质	分值
1	3	10^9	无	8
2	4			8
3	5			8
4	100	100	$T \leq 10$	10
5	500	500		10
6	3000			10
7	10^5			15
8	10^6	10^9	$b_i > 0$	5
9			a_i, b_i 单调不减	8
10			无	18