2026 深圳职业技术大学程序设计竞赛

2026-05-10

题目列表

ID	标题	时间限制	空间限制
A	简单的前后缀问题	2.0s	512MB
B	困难的树上 MEX 问题	2.0s	512MB
C	能见度	3.0s	1024MB
D	简单的填充问题	2.0s	512MB
E	简单的几何问题	2.0s	512MB
F	找索引	2.0s	512MB
G	富有挑战的图上问题	2.0s	512MB
H	简单的 RBS 问题	2.0s	512MB
I	?! 强强!?	2.0s	512MB
J	DFS 序对 (Easy Version)	2.0s	256MB
K	DFS 序对 (Hard Version)	3.0s	512MB
L	简单的区间问题	2.0s	512MB
M	困难的图论问题	1.0s	512MB

本次比赛共计 13 道正式题目

题目 A. 简单的前后缀问题

输入文件: stdin
输出文件: stdout
时间限制: 2 秒
内存限制: 512 MB

给定两个字符串 S,T，你需要判断 S 是否同时是 T 的前缀 † 且是 T 的后缀 ‡。

† 前缀：指从字符串第一个字符开始，连续截取到任意位置所组成的字符串。例如：`prefix` 的前缀有 `p`，`pre`，`prefix` 等。

‡ 后缀：指从字符串任意位置字符开始，连续截取到最后一位所组成的字符串。例如：`suffix` 的后缀有 `x`，`fix`，`ffix`，`suffix` 等。

输入

共输入两行：

第一行输入仅由大小写字母组成的字符串 $S(1 \leq |S| \leq 15)$。

第二行输入仅由大小写字母组成的字符串 $T(1 \leq |T| \leq 15)$。

其中 $|s|$ 表示字符串 s 的长度。

输出

如果字符串 S 为 T 的前缀且是 T 的后缀，则输出 `Yes`（大小写任意），否则输出 `No`（大小写任意）。

样例

stdin	stdout
Baka Kendieer	NO
Baka BakaCirnoBaka	YES

题目 B. 困难的树上 MEX 问题

输入文件: stdin
输出文件: stdout
时间限制: 2 秒
内存限制: 512 MB

给定一棵包含 n 个节点的有根树 T，根节点为 1。

每个节点 u 都有一个初始点权 w_u，且序列 $w_1, w_2, \ldots, w_n$ 是 $0, 1, \ldots, n-1$ 的一个排列 †。

对于树上的任意节点 u，定义其子树权值 $V(u)$ 为：

$$V(u) = \max_{v \in T_u} \Big(\mathrm{MEX}^{\ddagger}\big(\mathrm{path}(u, v)\big) \Big)$$

其中：T_u 表示以 u 为根的子树节点集合。$\mathrm{path}(u, v)$ 表示从节点 u 到节点 v 的简单路径上的所有节点权值的集合。你的任务是求出所有 $1 \le u \le n$ 对应的 $V(u)$。

‡ $\mathrm{MEX}(S)$：表示集合 S 中未出现的最小非负整数。例如：$\mathrm{MEX}(\{0, 1, 3\}) = 2$，$\mathrm{MEX}(\{1, 2\}) = 0$。

‡ 排列：由 $0, 1, \ldots, n-1$ 共 n 个不同整数组成的序列。每个数字在序列中恰好出现一次。例如：$(1, 2, 0, 3)$ 是排列，而 $(1, 2, 2, 0, 1), (1, 0, 4, 5), (0, 2)$ 均不是排列。

输入

第一行输入一个正整数 $T(1 \le T \le 10^4)$，表示测试组数。

接下来的 T 组测试中，对于每组测试，输入格式如下：

- 第一行输入一个正整数 $n(1 \le n \le 2 \times 10^5)$，表示树的节点数。
- 第二行输入 n 个非负整数，$w_1, w_2, \ldots, w_n(0 \le w_i < n)$，表示每个点的权值，保证给出的一定是一个排列。
- 接下来的 $n-1$ 行，每行输入两个正整数 $u, v(1 \le u, v \le n, u \ne v)$，表示一条边，且保证给出的边一定能形成一棵树。

此外，数据保证所有测试组 n 的总和不超过 2×10^5，保证给出的所有图一定是一棵树。

输出

对于每组测试组，共输出一行 n 个非负整数，表示 u 为根的子树下其权值大小。

样例

stdin	stdout
4 5 1 0 2 3 4 1 2 2 3 3 4 4 5 5 0 1 2 3 4 1 2 2 3 3 4 4 5 5 1 0 2 4 3 1 2 2 3 1 4 4 5 6 5 0 2 1 4 3 1 2 2 3 2 4 3 5 3 6	5 1 0 0 0 5 0 0 0 0 3 1 0 0 0 2 2 0 0 0 0

题目 C. 能见度

输入文件: stdin
输出文件: stdout
时间限制: 3 秒
内存限制: 1024 MB

给定一个 $n \times m$ 方格，其中坐标 (x, y) 的高度表示为 $h_{x,y}$，初始均为高度为 0，超出边界外均属于围墙。现在进行 q 次操作，每次操作进行以下操作中的一种：

1. 将坐标 (x_1, y_1) 到 (x_2, y_2) 之间所有坐标 $h_{i,j}$ 高度增加 k。
2. 查询坐标 (x, y) 十字范围内能看到 * 的空地格子个数（包括自身位置）。

$*$ 对于两个格子 $(x_1, y_1), (x_2, y_2)$，如果 (x_1, y_1) 能看到 (x_2, y_2)，则需要同时满足：

- $1 \le x_2 \le n$ 且 $1 \le y_2 \le m$
- $x_1 = x_2$ 或者 $y_1 = y_2$。
- $h_{x_1,y_1} \ge h_{x_2,y_2}$。
- (x_2, y_2) 四周至少一个格子是能被 (x_1, y_1) 看到。即 (x_2+1, y_2)、(x_2-1, y_2)、(x_2, y_2+1) 和 (x_2, y_2-1) 中至少有一个格子可以被 (x_1, y_1) 看到。

特别地，任何格子 (x, y) 都能看到其所在位置的格子，即 (x, y) 本身。

输入

第一行输入三个正整数 $n, m, q (1 \le n, m \le 100, 1 \le q \le 100)$，含义见题意。

接下来 q 行，每行首先输入一个正整数 $op (op \in \{1, 2\})$，表示操作编号，对于每种操作，输入格式如下：

1. 输入 6 个正整数 1 x_1 y_1 x_2 y_2 k $(1 \le x_1 \le x_2 \le n, 1 \le y_1 \le y_2 \le m, 1 \le k \le 10^4)$，表示操作 **1**。
2. 输入 3 个正整数 2 x y $(1 \le x \le n, 1 \le y \le m)$，表示操作 **2**。

输出

输出共一行，对于每次操作 **2**，输出一个非负整数，表示答案。

样例

stdin	stdout
5 6 9 1 1 1 5 2 1 1 1 5 2 6 2 1 5 1 5 6 4 1 1 3 1 5 1 1 3 4 3 4 76 2 2 2 2 2 3 2 2 5 2 3 4	7 4 8 10

注释

对于样例，可见网页上动图

题目 D. 简单的填充问题

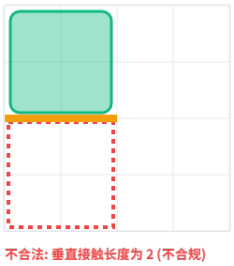
输入文件:	stdin
输出文件:	stdout
时间限制:	2 秒
内存限制:	512 MB

Kendieer 现在手头上有一个 $n \times m$ 的网格矩形，他觉得太空旷了，于是他决定塞一些 2×2 的正方形进入该矩形之中。

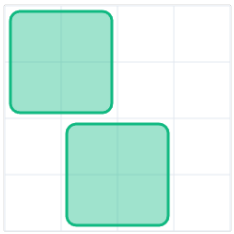
不过对于每一个被填入矩形的正方形而言，都需要满足：

- 每个 2×2 的正方形必须恰好覆盖网格中的 4 个完整小方格。
- 正方形不能超出 $n \times m$ 的矩形边界。
- 对于其他任意一个 2×2 的正方形，与当前正方形接触的长度不超过 1。

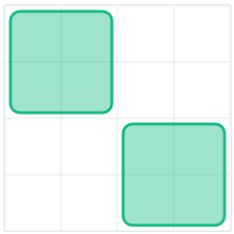
例如以下放置方式：



不合法



合法



合法

Duanhen 看到这个问题之后，觉得太简单了，于是他破坏了 k 个格子，使得这些格子均不可被正方形覆盖。

由于合法的填充方案太多了，Kendieer 多得数不过来，于是他向你请教一下一共有多少种本质不同 † 的合法放置方案，这个答案可能很大，你需要将答案对 998244353 取模。

†：两种方案被视为不同，当且仅当存在至少一个 2×2 的正方形位置，在一种方案中被放置而在另一种方案中未被放置。

输入

第一行输入两个正整数 $n, m (2 \le m \le n \le 12)$。

第二行输入一个非负整数 $k (0 \le k \le n \times m)$，表示被破坏的格子数量。

接下来 k 行，每行输入两个正整数 $x_i, y_i (1 \le x_i \le n, 1 \le y_i \le m)$，表示被破坏的格子坐标，数据保证坐标两两不同。

输出

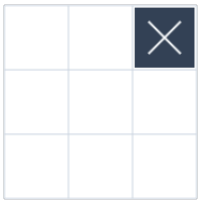
输出一个非负整数，表示答案。

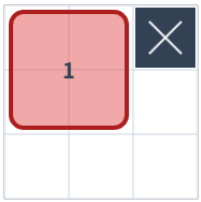
样例

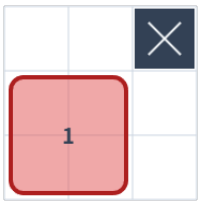
stdin	stdout
3 3 1 1 3	4
7 6 2 7 6 7 3	3388
12 12 1 7 6	567132745

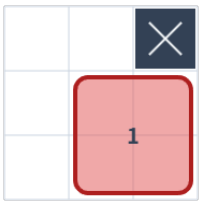
注释

对于样例 1，共有以下 4 种方案：（注意，不放置正方形也视作一种合法的放置方案）









题目 E. 简单的几何问题

输入文件: stdin
输出文件: stdout
时间限制: 2 秒
内存限制: 512 MB

Kendieer 有一个 $n \times m$ 的方格纸，初始时所有格子是白色的。

他涂黑了若干个格子，若我们将这些涂黑的格子视作一整个图形，他想知道这些格子所围成的图形的周长是多少。

输入

第一行输入一个正整数 $T(1 \le T \le 100)$，表示测试组数。

对于每组测试，输入格式如下：

- 第一行输入两个正整数 $n, m(1 \le n, m \le 1000, 1 \le n \times m \le 1000)$，含义见题意。
- 接下来 n 行，每行输入长度为 m 且仅由 `01` 组成的字符串，其中 `0` 表示空白格子，`1` 表示黑色格子。

数据保证所有测试点 $n \times m$ 的总和不超过 1000。

输出

对于每组测试，输出一个非负整数，表示答案。

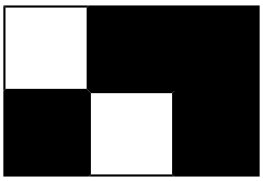
样例

stdin	stdout
3 2 3 011 101 2 2 00 00 4 4 0111 1101 1011 1110	12 0 24

注释

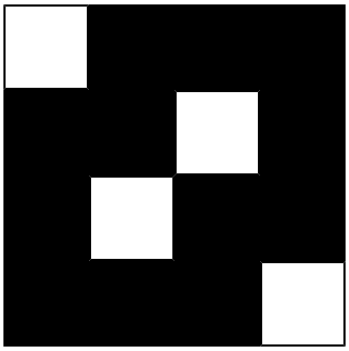
对于样例 1：

其图形如下，容易得出黑色部分形成的图形周长为 12。



对于样例 3：

其图形如下，容易得出黑色部分形成的图形周长为 24。



题目 F. 找索引

输入文件: stdin
输出文件: stdout
时间限制: 2 秒
内存限制: 512 MB

给定一个长度为 n 的序列 a，且这个序列保证单调不递减，即 $\forall 1 \le i < n, a_i \le a_{i+1}$。

现在进行 q 次询问，对于每次询问，求值为 x 在序列 a 中的索引，如果存在多个符合条件的索引任意一个即可，如果不存在则输出 -1。

输入

第一行输入两个正整数 $n, q (1 \le n, q \le 2 \times 10^5)$，表示序列大小和询问次数。

第二行输入 n 个正整数 $a_1, a_2, \ldots, a_n (1 \le a_i \le 10^9)$。

接下来 q 行，每行输入一个正整数 $x (1 \le x \le 10^9)$，表示询问。

输出

对于每次查询，如果存在对应索引则输出任意符合条件的索引；否则输出 -1。

如果存在多种合法解，输出其中任意一种即可。

样例

stdin	stdout
4 3 1 1 4 5 1 4 7	2 3 -1

题目 G. 富有挑战的图上问题

输入文件: stdin
输出文件: stdout
时间限制: 2 秒
内存限制: 512 MB

自从上次之后，Kendieer 领略了更多关于哈密顿路径的知识，现在他决定向你提出一个挑战：

给定一个由 n 个顶点组成的**有向无环图** G，求对于该图 G，共有多少条不同的哈密顿路径 *，这个结果可能很大，你需要将结果对 10^9+7 取模后输出。

* 哈密顿路径：对于一个图上的路径，如果该路径满足过图上所有点**有且仅有一次**，则称这个路径为哈密顿路径。

将图中的每条边视为唯一的对象（例如按照输入顺序编号为 $1,2,\ldots,m$）。两条路径不同，当且仅当它们使用的边的编号序列不同。

输入

第一行输入一个正整数 $T(1 \le T \le 10^4)$，表示测试数据组数。

接下来的 T 组数据，每组输入格式如下：

- 第一行输入两个正整数 $n,m(2 \le n \le 2\times 10^5, n-1 \le m \le 4\times 10^5)$，表示该图顶点个数和边的个数。
- 接下来的 m 行，每行输入 2 个正整数 $u,v(1 \le u,v \le n, u \ne v)$，表示一条 u 指向 v 的有向边。

此外，数据保证所有测试点 n 的总和不超过 2×10^5，m 的总和不超过 4×10^5，且保证给出的图一定是**有向无环连通图**。

输出

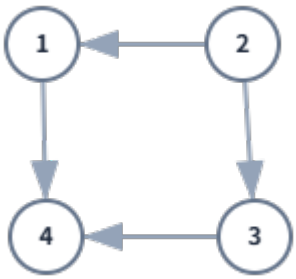
对于每组数据，输出一个非负整数，表示结果对 10^9+7 取模后的答案。

样例

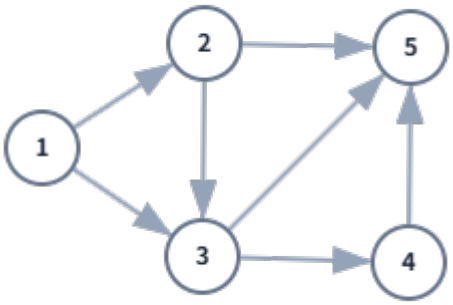
stdin	stdout
3 4 4 2 1 2 3 1 4 3 4 2 3 1 2 1 2 1 2 5 7 1 2 1 3 2 3 2 5 3 4 3 5 4 5	0 3 1

注释

对于第一组样例，其图如下：



对于第三组样例，其图如下：



该图仅存在一条哈密顿路径：$1 \to 2 \to 3 \to 4 \to 5$。

题目 H. 简单的 RBS 问题

输入文件: stdin
输出文件: stdout
时间限制: 2 秒
内存限制: 512 MB

给定一个长度为 n 的合法括号串 † S，现在可以进行任意次以下操作（包含零次）：

- 选择一个区间 $[l, r]$，且满足子串 ‡ $S[l \dots r]$ 是 RBS。
- 翻转该区间。即将 $s_l s_{l+1} \dots s_{r-1} s_r$ 变成 $s_r, s_{r-1} \dots s_{l+1} s_l$。
- 将该区间原本的 (变成)，) 变成 (。

例如：()(()) 选择区间 $[1, 6]$ 操作后变成 (())()，(()(())) 选择区间 $[2, 7]$ 操作后变成 ((())())。

求操作任意次括号串后，一共有多少种可能的合法括号串结果，这个答案可能很大，你需要输出答案对 998244353 取模后的结果。

† 合法括号串（RBS）：我们定义以下字符串为 RBS。

- 空字符串是一个 RBS；
- 如果 S 是一个 RBS，那么 (S) 也是一个 RBS；
- 如果 S 和 T 都是 RBS，那么 ST（即 S 和 T 拼接）也是一个 RBS。

例如：

- ()() 是 RBS。
- (()()) 是 RBS。
-)(()() 不是 RBS。

‡ 子串：字符串 S 的一个子串是由 S 中连续的一段字符组成的序列。具体地，对于字符串 $S = s_1 s_2 \dots s_n$，其子串 $S[i \dots j]$ $(1 \le i \le j \le n)$ 定义为字符串 $s_i s_{i+1} \dots s_j$。

输入

第一行输入一个正整数 $T (1 \le T \le 10^4)$，表示测试数据组数。

接下来的 T 组测试，对于每组测试，输入格式如下：

- 第一行输入一个正整数 $n (2 \le n \le 10^5)$，且保证 n 为偶数。
- 第二行输入一个长度为 n 的合法括号序列（RBS）S。

此外，数据保证所有测试 n 的总和不超过 10^5。

输出

对于每组数据，输出一个整数表示答案。

样例

stdin	stdout
6 6 (()()) 12 ()(()(()))() 12 (()())(()()) 12 (()())((())) 18 ()(()(()))(()(())) 20 (())(())()(()((()))))	1 6 1 2 12 24

题目 I. ?! 强强!?

输入文件:	`stdin`
输出文件:	`stdout`
时间限制:	2 秒
内存限制:	512 MB

Kendieer 根本就不强！所以他要变得强起来，他想到了一个办法。

他有一个长度为 n 序列 a，我们定义一个序列的能量为 $\sum_{i=1}^{n} a_i$，即这个序列所有元素的总和。

你可以对任意一个无前导零的数字执行任意次（包括 0 次）以下操作，其代价为累加：

- 将该数字任意一个数位从 x 修改到 y，其需要代价为 $|x-y|$，允许操作后的数字含有前导零。

例如：

- 从 123 修改成 223 代价为 1。
- 从 235 修改成 295 代价为 6。
- 从 235 修改成 325 代价为 2。

此外，我们定义最终能量为修改后的序列能量减去修改操作所需的代价。

现在，Kendieer 想最大化最终能量，从而显得他很强（其实并非强），请你输出这个序列可能的最大最终能量。

输入

第一行输入一个正整数 $T(1 \le T \le 10^4)$，表示测试数据组数。

接下来的 T 组数据中，每组数据输入如下：

- 第一行输入一个正整数 $n(1 \le n \le 2 \times 10^5)$，表示序列长度。
- 第二行输入一个 n 个正整数 $a_1, a_2, \ldots, a_n(1 \le a_i \le 10^9)$，表示序列。保证所有数字均无前导零。

此外，数据保证所有测试点 n 的总和不超过 2×10^5。

输出

对于每组测试，共输出一个整数，表示答案。

样例

stdin	stdout
3 6 1 1 4 5 1 4 9 9 9 8 2 4 4 3 5 3 3 233 1919810 998244353	16 47 1010000910

注释

对于第一组测试，不修改即可达到最大化最终能量。

对于第二组测试，一种可行的方案是将序列修改成 $[9,9,8,8,7,7,6,6,5]$，其代价为 18，可以证明不存在最终能量更优的操作方案。

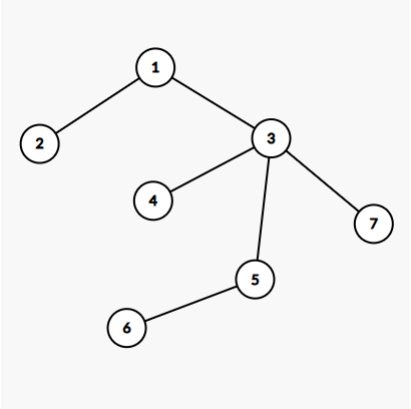
题目 J. DFS 序对 (Easy Version)

输入文件:	stdin
输出文件:	stdout
时间限制:	2 秒
内存限制:	512 MB

本题定义、输入格式、输出格式均与困难版本相同，在这个版本中，$n \le 2 \times 10^5$，但你只需构造出任意符合条件的树即可。

排列：长度为 n 的定义为由 $1 \sim n$ 以不同顺序组成的序列，且每个元素恰好出现一次。例如 $[2,1,3],[2,4,1,3]$ 都是排列，而 $[4,1,2]$ 不是排列，因为 3 没有出现，$[1,2,3,2]$ 不是排列，因为 2 出现了 2 次。

树的 dfs 序：对一棵树进行**深度优先遍历**（DFS），每当我们第一次访问某个节点时，就将该节点编号加入序列中。得到的访问顺序称为 DFS 序。例如：



对于给定的树，$[1,2,3,4,5,6,7],[1,3,4,7,5,6,2],[3,4,1,2,7,5,6]$ 均是这棵树能形成的 dfs 序。$[1,2,4,3,5,6,7]$ 不是这棵树的 dfs 序，因为 2 号点到 4 号点存在未访问的 3 号点，$[1,2,3,4,5,7,6]$ 不是这棵树的 dfs 序，因为子树 5 不满足 DFS 遍历性质。

现在，给定 2 个 $1 \sim n$ 组成的排列，现在要求你构造一棵树，使得这棵树**包含**这两种 dfs 序，可以证明这样的树总是存在的。

输入

第一行输入一个正整数 $T(1 \le T \le 10^4)$，表示测试组数。

接下来每组测试中：

- 第一行输入一个正整数 $n(3 \le n \le 2 \times 10^5)$
- 第二行输入一个长度为 n 的排列 $p_1, p_2, \cdots, p_n$，表示树的其中一种 dfs 序。
- 第三行输入一个长度为 n 的排列 $q_1, q_2, \cdots, q_n$，表示树的另一种 dfs 序。
- 此外，保证 $p_1 = 1, q_1 = 1$。

此外，数据保证所有测试点的 n 之和不超过 2×10^5。

输出

对于每组数据，输出共 $n-1$ 行，使其形成一棵树，每行输出 2 个正整数 $u, v (1 \le u, v \le n, u \ne v)$，表示一条边。

如果存在多种合法解，输出其中任意一种即可。

样例

stdin	stdout
3 6 1 2 3 6 5 4 1 2 3 4 5 6 8 1 2 8 5 7 4 6 3 1 6 3 4 5 7 2 8 7 1 2 3 4 5 6 7 1 3 4 7 5 6 2	1 2 2 3 3 4 3 5 3 6 1 2 2 8 1 5 5 7 1 4 1 6 6 3 1 2 1 3 3 4 3 5 5 6 3 7

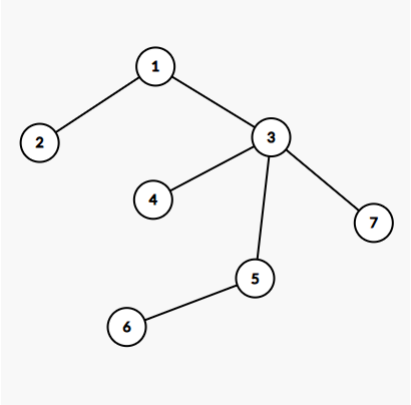
题目 K. DFS 序对 (Hard Version)

输入文件: stdin
输出文件: stdout
时间限制: 3 秒
内存限制: 512 MB

本题定义、输入格式、输出格式均与简单版本相同，在这个版本中，$n \le 5000$，但你需要构造出符合条件的树且每个节点深度总和最大化的方案。

排列：长度为 n 的定义为由 $1 \sim n$ 以不同顺序组成的序列，且每个元素恰好出现一次。例如 $[2,1,3],[2,4,1,3]$ 都是排列，而 $[4,1,2]$ 不是排列，因为 3 没有出现，$[1,2,3,2]$ 不是排列，因为 2 出现了 2 次。

树的 dfs 序：对一棵树进行**深度优先遍历**（DFS），每当我们第一次访问某个节点时，就将该节点编号加入序列中。得到的访问顺序称为 DFS 序。例如：



对于给定的树，$[1,2,3,4,5,6,7],[1,3,4,7,5,6,2],[3,4,1,2,7,5,6]$ 均是这棵树能形成的 dfs 序。$[1,2,4,3,5,6,7]$ 不是这棵树的 dfs 序，因为 2 号点到 4 号点存在未访问的 3 号点，$[1,2,3,4,5,7,6]$ 不是这棵树的 dfs 序，因为子树 5 不满足 DFS 遍历性质。

树的深度：我们定义根节点的深度为 1。对于其他任意节点 u，其深度为 u 的父节点的深度加 1。

现在，给定 2 个 $1 \sim n$ 组成的排列，现在要求你构造**包含**这两种 dfs 序的树，且使得每个节点深度总和**最大化**。可以证明无论序列如何，总是存在对应的树。

输入

第一行输入一个正整数 $T(1 \le T \le 10^3)$，表示测试组数。

接下来每组测试中：

- 第一行输入一个正整数 $n(3 \le n \le 5000)$
- 第二行输入一个长度为 n 的排列 $p_1, p_2, \cdots, p_n$，表示树的其中一种 dfs 序。
- 第三行输入一个长度为 n 的排列 $q_1, q_2, \cdots, q_n$，表示树的另一种 dfs 序。
- 此外，保证 $p_1 = 1, q_1 = 1$。

此外，数据保证所有测试点的 n 之和不超过 5000。

输出

对于每组数据，输出共 $n-1$ 行，使其形成一棵树，每行输出 2 个正整数 $u, v (1 \le u, v \le n, u \ne v)$，表示一条边。

如果存在多种合法解，输出其中任意一种即可。

样例

stdin	stdout
3 6 1 2 3 6 5 4 1 2 3 4 5 6 8 1 2 8 5 7 4 6 3 1 6 3 4 5 7 2 8 7 1 2 3 4 5 6 7 1 3 4 7 5 6 2	1 2 2 3 3 6 3 5 3 4 1 2 2 8 1 5 5 7 1 4 1 6 6 3 1 2 1 3 3 4 4 5 5 6 4 7

题目 L. 简单的区间问题

输入文件: stdin
输出文件: stdout
时间限制: 2 秒
内存限制: 512 MB

给定一个长度为 n 的序列 a，我们定义一个子区间 † $[l,r]$ 的权值 $f(a[l\dots r])=\max_{i=l}^{r}a_i+\min_{i=l}^{r}a_i$，即区间最大值与区间最小值之和。

你需要求出所有子区间权值的总和，即求 $\sum_{i=1}^{n}\sum_{j=i}^{n}f(a[i\dots j])$。

† 子区间：一个序列 a 的子区间 $[l,r]$ $(1\le l\le r\le n)$ 定义为原序列中连续的一段元素：$a_l,a_{l+1},\dots,a_r$，记作 $a[l\dots r]$ 的总和值。

输入

第一行输入一个正整数 $T(1\le T\le 10^4)$，表示数据组数。

接下来的 T 组数据中，每组数据输入如下：

- 第一行输入一个正整数 $n(1\le n\le 2\times 10^5)$，表示序列长度。
- 第二行输入 n 个正整数 $a_1,a_2,\dots,a_n(1\le a_i\le 10^8)$。

此外，数据保证所有测试组 n 的总和不超过 2×10^5。

输出

对于每组测试，共输出一行一个整数，表示答案。

样例

stdin	stdout
6 3 5 3 7 6 1 5 4 2 8 5 6 1 4 2 6 3 3 9 9 9 8 2 4 4 3 5 3 7 12 5 3 10 4 18 11 12 1 1 4 5 1 4 1 9 1 9 8 10	58 181 147 448 505 710

注释

对于第一组测试：

- $f(a[1]) = 5 + 5 = 10$，5 同时是该区间最大值和最小值。
- $f(a[2]) = 3 + 3 = 6$，3 同时是该区间最大值和最小值。
- $f(a[3]) = 7 + 7 = 14$，7 同时是该区间最大值和最小值。
- $f(a[1 \dots 2]) = 5 + 3 = 8$，5 是该区间的最大值，3 是该区间的最小值。
- $f(a[2 \dots 3]) = 7 + 3 = 10$，7 是该区间的最大值，3 是该区间的最小值。
- $f(a[1 \dots 3]) = 7 + 3 = 10$，7 是该区间的最大值，3 是该区间的最小值。

题目 M. 困难的图论问题

输入文件: stdin
输出文件: stdout
时间限制: 1 秒
内存限制: 512 MB

Kendieer 遇到了一个困难的图论问题，这个问题同时涉及到了完全图和二分图概念问题，这令他这个图论新手头晕目眩。

我们定义如果数量为 n 个顶点的完全图 † 是二分图 ‡，则这个图则被称之为完全好图。

现在你需要求对于顶点个数 $l \le n \le r$ 所有 n 的情况下，其中完全好图的个数。

† 完全图：对于一个由 m 个顶点组成的完全图，满足任意两个不同顶点 u, v 之间都存在边 (u, v)。

‡ 二分图：对于一个图，将所有点划分至两个集合，若存在一种划分方式，所有边都满足 (u, v) 连接的两个顶点 u, v 不属于同一个集合，则这样的图称之为二分图。

输入

第一行输入一个正整数 $T(1 \le T \le 10^4)$，表示测试组数。

接下来的 T 组数据中，每组测试输入两个正整数 $l, r(1 \le l \le r \le 10^9)$，含义见题意。

输出

对于每组测试，共输出一个非负整数，表示答案。

样例

stdin	stdout
3 2 3 5 5 11 13	1 0 0