

CSP2023模拟赛

| 题目名称 | 探险家 | 最近公共祖先 | 跳跃 | 翻转 |
|-----------|--------------|---------|----------|----------|
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |
| 可执行文件名 | explorer | lca | jump | flip |
| 输入文件名 | explorer.in | lca.in | jump.in | flip.in |
| 输出文件名 | explorer.out | lca.out | jump.out | flip.out |
| 每个测试点时限 | 2.0 秒 | 2.0 秒 | 2.0 秒 | 2.0 秒 |
| 内存限 | 512 MiB | 512 MiB | 512 MiB | 512 MiB |
| 子任务/测试点数目 | 20 | 20 | 20 | 4 |
| 是否等分 | 是 | 是 | 是 | 否 |

提交源文件程序名

| | | | | |
|---------|--------------|---------|----------|----------|
| 对于C++语言 | explorer.cpp | lca.cpp | jump.cpp | flip.cpp |
|---------|--------------|---------|----------|----------|

编译选项

| | |
|---------|--------------------|
| 对于C++语言 | -lm -O2 -std=c++17 |
|---------|--------------------|

注意事项(请仔细阅读)

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++中函数main()的返回类型必须是int，程序正常结束时返回值必须是0。
3. 选手提交的程序代码文件请在**个人目录下以及子文件夹内各放一份**。
4. 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）。
5. 选手提交的程序源文件必须不大于100KB。
6. 程序可使用的栈空间内存限制于题目的内存限制一直。
7. 使用std::deque等STL容器时，请注意其内存空间消耗。
8. 评测时采用的机器配置为 AMD Ryzen 7 5800H with Radeon Graphics，内存16GiB。上述时限以此配置为准。
9. 评测在Windows 10下进行，使用LemonLine进行评测。

探险家 (explorer)

探险家小 M 来到了一片新的大陆。这片大陆有 $2n$ 座城市，第 i 座城市和第 $i + 1$ 座之间有一条双向道路。特殊地，第 $2n$ 座城市和第 1 座城市之间也有一条道路。换句话说，这片大陆上的 $2n$ 座城市连成了一个环。

经过细致的调查，小 M 发现，这片大陆内部有 n 个黑暗势力，每个黑暗势力恰好控制了这片大陆上的两个城市，且没有两个黑暗势力会控制一个相同的城市。具体来说，第 i 个黑暗势力控制了第 a_i, b_i 座城市。

一位叫做以纳的神秘人找到小 M ，并告诉她，自己要收买这篇大陆的三个黑暗势力，并将这三个黑暗势力控制的城市作为自己的根据地。同时，以纳提出了一定的要求。定义黑暗势力 i 的强度为，以纳收买完三个黑暗势力后，从第 a_i 座城市到第 b_i 座城市至少经过的根据地数量。如果收买的黑暗势力中，某些黑暗势力的强度太大，那么可能会引发叛变，因此，以纳希望这三个黑暗势力的强度相等。

选手目录下的 `explorer/sample.png` 展示了一种 $n = 4$ 时的黑暗势力分布。如果以纳收买了红、黄、绿三个颜色的黑色势力，那么红色黑暗势力的强度为 2 ，绿色黑暗势力的强度为 4 ，黄色黑暗势力的强度为 2 ，这不是以纳所希望的。

以纳需要小 M 帮忙求出，有多少种**符合要求的**选择这三个黑暗势力的方法，并保证，事成之后分小 M 价值 5 RMB 的财产，让她可以买一瓶水喝。和往常一样，因为小 M 非常可爱，所以你需要帮助她。

【输入格式】

第一行一个正整数 n 。
接下来 n 行，每行两个正整数 a_i, b_i 。

【输出格式】

一行一个非负整数表示答案。

【输入输出样例1】

| explorer.in | explorer.out |
|-------------------------------|--------------|
| 4 1 2 3 6 4 8 5 7 | 0 |

【输入输出样例1说明】

该样例对应题目描述中的图片。容易发现，没有任何合法的方案。

【输入输出样例2】

| explorer.in | explorer.out |
|---------------------------------------|--------------|
| 5 1 8 2 5 3 9 4 10 6 7 | 3 |

【输入输出样例2说明】

合法的三种方案分别为：选择黑暗势力 1, 2, 5, 三个黑暗势力的强度都是 2；选择黑暗势力 2, 3, 4, 三个黑暗势力的强度都是 4；选择黑暗势力 1, 3, 4, 三个黑暗势力的强度都是 4。

【数据规模与约定】

- 对于前 20% 的测试数据， $n \leq 500$ 。
- 对于前 40% 的测试数据， $n \leq 2000$ 。
- 对于前 60% 的测试数据， $n \leq 10^5$ 。
- 另有 10% 的测试数据，满足 $a_i = 2i - 1, b_i = 2i$

对于所有测试数据， $1 \leq n \leq 5 \times 10^5, 1 \leq a_i, b_i \leq 2n$ 。

最近公共祖先 (lca)

【题目描述】

给定一棵无根树，节点从 1 开始标号，求以每个节点为根时， $\sum_{1 \leq i \leq n} \sum_{1 \leq j \leq n} lca(i, j)$ 的值。

其中， $lca(i, j)$ 表示点 i 和点 j 的最近公共祖先的编号。

不要求 $i < j$ ，即对于 $i \geq j$ 也要统计答案。

【输入格式】

第一行一个正整数 n 。

第 2 至 n 行，每行两个数，第 $i + 1$ 行的两个数 u_i, v_i 表示存在一条 (u_i, v_i) 之间的无向边。

保证输入形成一棵树。

【输出格式】

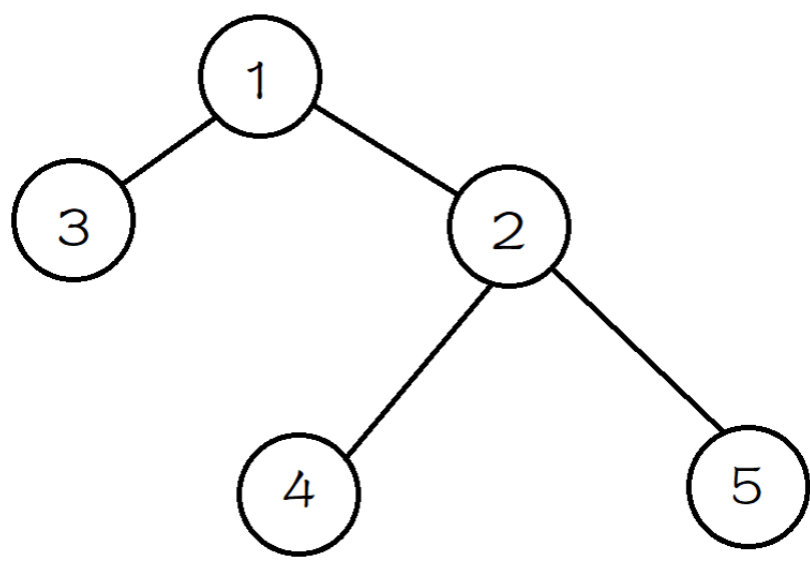
一行，共 n 个数，第 x 个数表示以 x 为根时 $\sum_{i \leq i \leq n} \sum_{1 \leq j \leq n} lca(i, j)$ 的值。

【输入输出样例1】

| lca.in | lca.out |
|-------------------------------|----------------|
| 5 1 2 1 3 2 4 2 5 | 41 53 57 69 77 |

【输入输出样例 1 说明】

由于输入的树如图所示



【数据规模与约定】

- 对于前 10% 的测试数据, $n \leq 50$
- 对于前 20% 的测试数据, $n \leq 300$
- 对于前 40% 的测试数据, $n \leq 4000$
- 另有 15% 的测试数据, 满足 $u_i = i, v_i = i + 1$
- 另有 15% 的测试数据, 满足 $u_i = i + 1, v_i = \lfloor \frac{i+1}{2} \rfloor$

对于所有的测试数据, 保证 $2 \leq n \leq 5 \times 10^5$, 保证输入的是一棵树。

跳跃 (jump)

【题目描述】

给定一个数列 a 表示一个共 $n + 1$ 格的棋盘，方格编号从 0 开始。其中 $a_0 = \infty, a_1 \dots a_n$ 由输入给定。对于 $1 \leq x \leq n$ 的正整数。定义棋子从 x 处开始的跳跃操作为：找到最大的 y 满足 $y < x, a_y > a_x$ 将棋子移动至 y 处。可以证明这样的 y 必然存在。

对于 1 至 n 的每一个格子询问：如果一枚棋子从该格开始不断跳跃，直至位于第 0 格，求最终该棋子的跳跃次数。

询问之间相互独立。

【输入格式】

第一行一个正整数 n 。

第二行共 n 个正整数，依次为 a_1, a_2, \dots, a_n 。

【输出格式】

一行， n 个正整数，第 i 个表示位于方格 i 的棋子最终跳跃的次数。

【输入输出样例1】

| jump.in | jump.out |
|----------------|-----------|
| 5 1 2 3 4 5 | 1 1 1 1 1 |

【输入输出样例2】

| jump.in | jump.out |
|---------------------|---------------|
| 7 1 9 1 9 8 1 10 | 1 1 2 1 2 3 1 |

【数据规模与约定】

- 对于前 10% 的测试数据，满足 $n \leq 300$ 。
- 对于前 40% 的测试数据，满足 $n \leq 3000$ 。
- 另有 15% 的测试数据，满足 $1 \leq a_i \leq 2$ 。
- 另有 20% 的测试数据，满足 a_i 在数据范围内随机生成。

对于所有的测试数据，保证 $1 \leq n \leq 5 \times 10^5, 1 \leq a_i \leq 10^9$ 。

翻转 (flip)

【题目描述】

你有一个以 1 开头的长度为 $2n - 1$ 的 01 交替串。你每次可以选择一个以 1 为开头结尾的非空 01 交替子串翻转($0 \rightarrow 1, 1 \rightarrow 0$)，问有多少种翻转 k 次的方案。答案对 998244353 取模。

【输入格式】

一行两个正整数 n, k 。

【输出格式】

一行一个整数，表示方案对 998244353 取模的值。

【输入输出样例1】

| flip.in | flip.out |
|---------|----------|
| 3 2 | 15 |

【输入输出样例1说明】

一开始字符串为 10101，第一次操作可以操作的子串位置有 $[1, 1], [3, 3], [5, 5], [1, 3], [3, 5], [1, 5]$ 。

- 当第一次操作 $[1, 1]$ 时，原字符串变为 00101，第二次操作共有 3 种方案；
- 当第一次操作 $[3, 3]$ 时，原字符串变为 10001，第二次操作共有 2 种方案（只有 $[1, 1]$ 和 $[5, 5]$ ，因为 10001 并不是 01 交替子串）；
- 当第一次操作 $[5, 5]$ 时，原字符串变为 10100，第二次操作共有 3 种方案；
- 当第一次操作 $[1, 3]$ 时，原字符串变为 01001，第二次操作共有 2 种方案；
- 当第一次操作 $[3, 5]$ 时，原字符串变为 10010，第二次操作共有 2 种方案；
- 当第一次操作 $[1, 5]$ 时，原字符串变为 01010，第二次操作共有 3 种方案；

所以共有 15 种方案。

【数据规模与约定】

本题采用捆版测试。对于所有测试数据，满足 $1 \leq k \leq n \leq 10^6$ 。

| 子任务编号 | $n \leq$ | 特殊限制 | 子任务分值 |
|-------|----------|---------|-------|
| 1 | 10 | | 15 |
| 2 | 300 | | 25 |
| 3 | 10^6 | $n = k$ | 30 |
| 4 | 10^6 | | 30 |