NOIP 2024 模拟赛

NOIP 2024 Simulation

CJ 联合出题组

2024 年 11 月 15 日

题目名称	花坛	伊斯贝斯	虫洞	故事结局
可执行文件名	`flower`	`espace`	`wormhole`	`story`
输入文件名	`flower.in`	`espace.in`	`wormhole.in`	`story.in`
输出文件名	`flower.out`	`espace.out`	`wormhole.out`	`story.out`
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB	512MB
测试点数目	10	10	13	10
测试点是否等分	是	是	否	否

提交源程序文件名

对于 C++ 语言	`flower.cpp`	`espace.cpp`	`wormhole.cpp`	`story.cpp`

编译选项

对于 C++ 语言	`-lm -O2`	`-lm -O2`	`-lm -O2`	`-lm -O2`

【注意事项（请仔细阅读）】

1. 选手提交的源程序请**直接放在个人目录下**，无需建立子文件夹；
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
4. **对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理**。
5. 若无特殊说明，结果比较方式为**忽略行末空格、文末回车后的全文比较**。。
6. 程序可使用的栈空间大小与该题内存空间限制一致。

7. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
8. 若无特殊说明，每道题的**代码大小限制为 100KB**。
9. 若无特殊说明，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔。
10. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 scanf 函数）避免出错。
11. 直接复制 PDF 题面中的跨页样例，数据将带有页眉页脚，建议选手直接使用对应目录下的样例文件进行测试。
12. 使用 std::deque 等 STL 容器时，请注意其内存空间消耗。
13. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外**不允许在程序中手动开启其他编译选项**，一经发现，本题成绩以 0 分处理。

花坛（flower）

【题目描述】

小 N 有一个花坛，花坛从左到右依次排列了 n 朵花，其中第 i 朵花的美丽值为 a_i，每一朵花的美丽值都是 1 到 n 之间的正整数，且所有花的美丽值都互不相同。

小 N 定义一个花坛是美的，当且仅当对于 $1 \le i < j \le n$，均满足 $\frac{a_i}{a_j} \le \frac{j}{i}$，为了让花坛变为美的，小 N 每次可以选择一个满足 $1 \le i \le n-1$ 的正整数 i，并交换第 i 朵花与第 $i+1$ 朵花的美丽值。

现在他想知道至少要进行多少次交换操作，才能使得这个花坛是美的。

【输入格式】

从文件 ***flower.in*** 中读入数据。

输入的第一行包含一个正整数 n，表示花朵的个数。

接下来的一行包含 n 个正整数，其中第 i 个正整数表示第 i 朵花的美丽值。

【输出格式】

输出到文件 ***flower.out*** 中。

输出一行一个整数，表示至少要进行多少次邻项交换操作，才能使得这个花坛是美的。

【样例 1 输出】

```
4
1 4 2 3
```

【样例 1 输出】

```
1
```

【样例 1 解释】

对于初始花坛，其中当 $i = 2, j = 3$ 时不满足 $\frac{a_2}{a_3} \le \frac{3}{2}$，在交换编号 $2, 3$ 的花坛变为 $1, 2, 4, 3$，不难验证其满足条件。

【样例输入 2】

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

【样例输出 2】

```
7
```

【样例输入/输出 3】

见附件中的 ***flower3.in/ans***，该样例满足测试点 5 的限制。

【样例输入/输出 4】

见附件中的 ***flower4.in/ans***，该样例满足测试点 6 的限制。

【样例输入/输出 5】

见附件中的 ***flower5.in/ans***，该样例满足测试点 8 的限制。

【测试点约束】

对于所有测试点均满足：$1 \le n \le 10^6$，a_i 为一个大小为 n 的排列。
每个测试点的具体限制见下表：

测试点编号	$n \le$	特殊性质
1	1	A
2	5	A
3	5	无
4	20	A
5	20	无
6 ∼ 7	2000	A
8	2000	无
9	10^6	A
10	10^6	无

特殊性质 A: 对于所有满足 $1 \le i \le n-1$ 的奇数 i 均有 $a_i > a_{i+1}$。

伊斯贝斯（espace）

【题目描述】

公元 2202 年，伊斯贝斯大军入侵欧艾大陆，带来了前所未有的灾难。

作为欧艾大陆的守护者小 O，你决定对伊斯贝斯大军进行顽强的抵抗，经过 998244353 毫秒的战斗后，这场战争终于得以平息。

然而，在经过长期的观察后，你发现欧艾大陆仍然存在伊斯贝斯派来的间谍，为了防止战火重燃，你决定在暗中进行调查。

小 O 认为伊斯贝斯势力具有三种力量：喵力，卒力，独力，且它们会以一定的顺序进行排列，且由于这三股力量之间存在潜移默化的联系，它们之间可以相互转化。具体的，如果相邻的三个力量从左到右依次为喵力，卒力和独力，则可以将其分别变为卒力，独力和喵力；如果相邻的三个力量从左到右依次为卒力，独力和喵力，则可以将其分别变为独力，喵力和卒力；如果相邻的三个力量从左到右依次为独力，喵力和卒力，则可以将其分别变为喵力，卒力和独力。

现在小 O 得到了了一个由伊斯贝斯势力的力量组成的字符串 S (其中在字符串中，卒力代表 A，独力代表 B，喵力代表 C)，而小 O 认为一个欧艾公民是伊斯贝斯势力的当且仅当其力量所构成的字符串 T 可以通过若干次转化变为 S，现在他需要调查 k 名欧艾公民，并需要判断他是否属于伊斯贝斯势力。

这么难的问题小 O 当然不会做啦，于是他找到了你。

形式化的题意：有一个长为 n 的字符串 S，k 组询问，每次给出一个长为 n 的字符串 T，可以进行以下操作，每次可以选择一个数 i 满足 $1 \le i \le n-2$ 且满足 S_i, S_{i+1}, S_{i+2} 组成的字符串等于 ABC 或 BCA 或 CAB，然后将替换为 ABC, BCA, CAB 这三个字符串中的任意一个，判定是否能将 T 通过有限多次操作变为 S。

【输入格式】

从文件 ***espace.in*** 中读入数据。

输入的第一行包含一个正整数 n，表示力量构成的字符串的长度。

第二行包含一个长度为 n 的字符串 S，表示由伊斯贝斯势力的力量组成的字符串 S。

第三行包含一个正整数 k，表示小 O 需要对 k 个欧艾公民进行检验。

接下来的 k 行每行一个长为 n 的字符串 T，表示该公民的力量组成的字符串 T。

【输出格式】

输出到文件 ***espace.out*** 中。

输出共 k 行，如果根据小 O 的判定方式该公民属于伊斯贝斯势力则输出 yes，否则输出 no。

【样例输入 1】

```
5
ABCBC
3
ABCBC
BCBCA
BACBC
```

【样例输出 1】

```
yes
yes
no
```

【样例 1 解释】

对于第一位公民，有 $S = T$，故其属于伊斯贝斯势力。

对于第二位公民，可以先将后三个力量卒力，独力和喵力分别变为独力，喵力和卒力后再变为喵力，卒力和独力，前三个力量同样从卒力，独力和喵力分别变为独力，喵力和卒力后再变为喵力，卒力和独力，即 BCBCA->BCCAB->BCABC->CABBC->ABCBC，所以其也属于伊斯贝斯势力。

对于第三位公民，可以证明其所构成的字符串 T 不存在任何方案将其变为 S，故其不属于伊斯贝斯势力。

【样例输入/输出 2】

见附件中的 ***espace2.in/ans***，该样例满足测试点 2 的限制。

【样例输入/输出 3】

见附件中的 ***espace3.in/ans***，该样例满足测试点 4 的限制。

【样例输入/输出 4】

见附件中的 ***espace4.in/ans***，该样例满足测试点 7 的限制。

【测试点约束】

对于所有测试点均满足：$1 \le n \le 10^5, 1 \le nk \le 5 \times 10^6$。

每个测试点的具体限制见下表：

测试点编号	$n =$	$k =$	特殊性质
1	3	100000	A
2	7	100000	无
3	20	250000	无
4	50	100000	A
5	300	15000	无
6	1000	5000	A
7	3000	1500	无
8	10000	500	无
9	50000	100	无
10	100000	50	无

特殊性质 A: 保证 $S_1 = A, S_2 = B, S_3 = C$，且对于 $3 \le i \le n$ 均满足 $S_i = S_{i-2}$。

本题将会按照正确的行数进行给分，如果你在一个测试点中答对了 c 行，你将获得 $\lfloor 10 \times (\frac{c}{k})^{3.5} \rfloor$ 分，注意如果存在一组询问你输出的字符串不为 yes 或 no，你将在该测试点获得 0 分。

在下发文件中提供了 spj.cpp，选手可以自行使用。

虫洞（wormhole）

【题目描述】

小 I 来到了一个古老的星系，这里交错着分布着 n 个星球，星球之间有 m 条单向虫洞相接，且通过这些虫洞一定可以从 1 号星球到达其他任何一个星球，虫洞之间不存在闭合回路。

在这些虫洞中有 $n-1$ 条虫洞是关键的，这些虫洞掌控着星系交通的命脉，具体的，关键的虫洞可以由如下方式确定下来：

首先，令集合 S 为仅包含 1 号星球的集合，并将当前星球 x 设置为 1 号星球。

然后，选择一条以当前城市 x 为出发点、一个不在集合 S 内的城市 y 为终点的虫洞，将该虫洞标记为优先级高的虫洞，将城市 y 加入 S 集合，并将虫洞 y 的父亲设置为当前虫洞 x。

最后，将当前虫洞 x 修改为虫洞 y，如果虫洞 y 的选择有多个，则选择虫洞编号最小的一个；如果没有可以选择的虫洞 y，则将当前虫洞 x 修改为 x 的父亲。

重复上述操作，直到所有虫洞被加入 S 集合时停止。

可以发现，上述步骤恰好可以选出关键的 $n-1$ 条虫洞，并且这些关键的虫洞可以构成一个以 1 号星球为根的树型结构。

由于外星系势力的入侵，一些虫洞可能会因为受损而无法正常通行，现在小 I 想要知道，如果在关键虫洞所组成的树形结构上，a 号星球到 b 号星球 (保证在树形结构上 a 是 b 的祖先，且 $a \neq b$) 的最短路径上的所有虫洞都因为受损而无法正常通行时，从 1 号星球出发有多少个 b 号星球的子树中的星球无法到达，并希望精通算法的你能够告诉他答案。

由于这仅仅只是思想实验，小 I 会给出 q 组询问。

形式化的题意：有一个 n 个点 m 条边的 DAG，并且按照规则得到了其一个 dfs 树 T，q 组询问，每次查询如果将 T 上 a 到 b 这条祖先到后代的链上的边断掉后，1 将会有多少个 b 子树内的点将会到不了。

【输入格式】

从文件 ***wormhole.in*** 中读入数据。

第一行包含三个正整数 c, n, m, q，表示测试点编号 (特别的，样例中的 c 表示该样例的满足的限制与第 c 个测试点所满足的限制相同)，星球的个数，虫洞的个数与小 I 询问的个数。

接下来 m 行，每行包含两个正整数 x, y，表示存在一条 x 到 y 的虫洞，保证过这些虫洞一定满足可以从 1 号星球到达其他任何一个星球，且虫洞之间不存在闭合回路，且

同一条虫洞不会出现大于等于两次。

接下来 q 行，每行包含两个正整数 a, b，表示小 I 在给出的第 i 组询问中认为在关键虫洞所组成的树形结构上 a 号星球到 b 号星球的最短路径上的虫洞因为受损而无法正常通行。

【输出格式】

输出到文件 ***wormhole.out*** 中。

输出 q 行，每行一个整数，表示当 a 号星球到 b 号星球的最短路径上的所有虫洞都因为受损而无法正常通行时，从 1 号星球出发有多少个 b 号星球的子树中的星球无法到达。

【样例输入 1】

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

【样例输出 1】

```
0
1
```

【样例 1 解释】

共有 4 个星球和 4 条虫洞 $(1,2),(1,3),(2,4),(3,4)$，其中关键的虫洞为 $(1,2),(1,3),(2,4)$。

对于第一组询问，虫洞 $(1,2),(2,4)$ 将会无法正常通行，此时 1 号星球仍然可以通过 $(1,3),(3,4)$ 到达 4 号星球，4 子树内无法到达的星球数量为 0。

对于第二组询问，虫洞 $(1,2)$ 将会无法正常通行，此时 1 号星球可以通过 $(1,3),(3,4)$ 到达 4 号星球。

【样例输入 2】

```
1 6 8 5
1 6
6 3
3 4
3 2
6 5
1 2
2 5
4 5
3 4
1 3
1 2
1 4
1 5
```

【样例输出 2】

```
1
2
0
1
0
```

【样例输入/输出 3】

见附件中的 ***wormhole3.in/ans***，该样例满足测试点 2 的限制。

【样例输入/输出 4】

见附件中的 ***wormhole4.in/ans***，该样例满足测试点 5 的限制。

【样例输入/输出 5】

见附件中的 ***wormhole5.in/ans***，该样例满足测试点 7 的限制。

【样例输入/输出 6】

见附件中的 ***wormhole6.in/ans***，该样例满足测试点 8 的限制。

【样例输入/输出 7】

见附件中的 ***wormhole7.in/ans***，该样例满足测试点 10 的限制。

【测试点约束】

对于所有测试点均满足：$1 \le n, m, q \le 2 \times 10^5$，给出的 m 条虫洞一定可以满足从 1 号星球到达其他任何一个星球，且虫洞之间不存在闭合回路，同一条虫洞不会出现大于等于两次。

每个测试点的具体限制见下表：

测试点编号	$n \le$	$m - n \le$	$q \le$	特殊性质	测试点分值
1	10	10	10	无	6
$2 \sim 3$	2000	2000	2000	无	7
4	2×10^5	-1	2×10^5	无	6
5	2×10^5	0	2×10^5	A	6
6	2×10^5	0	2×10^5	B	7
7	2×10^5	0	2×10^5	无	7
$8 \sim 9$	2×10^5	2×10^5	2×10^5	A	9
$10 \sim 11$	2×10^5	2×10^5	2×10^5	B	9
$12 \sim 13$	2×10^5	2×10^5	2×10^5	无	9

特殊性质 A：保证 a 是 b 的父亲。

特殊性质 B：保证 $a = 1$。

故事结局（story）

【题目描述】

给定 n 和 y 与一个长为 n 的整数序列 A，保证序列 A 是单调不降或单调不增的。

现在小 P 想要生成一张 n 个点的有向图 G，其中 i 存在向 j 的连边当且仅当 $a_i \geq j$。

小 P 定义 G 的一个边子集 E 是好的当且仅当其入度和出度都不超过 1，且定义其权值为 $y^{W(T)}$，其中 $W(T)$ 表示 T 的环数 (这里认为 $0^0 = 1$)。

现在小 P 迫切想要知道对于每一个整数 $0 \leq k \leq n$，大小为 k 的合法边子集的权值和对 998244353 取模的结果。

因为所有的故事又回到了起点，一切又将重新开始。

【输入格式】

从文件 ***story.in*** 中读入数据。

第一行包含三个正整数 c, n, y，其中第一个数表示测试点编号 (特别的，样例中的 c 表示该样例的满足的限制与第 c 个测试点所满足的限制相同)，第二个数表示序列的长度，第三个数表示好边集权值的底数。

接下来一行 n 个整数，第 i 个数为表示 a_i。

【输出格式】

输出到文件 ***story.out*** 中。

输出一行 $n+1$ 个整数，其中第 i 个整数代表大小为 $i-1$ 的合法边子集的权值和对 998244353 取模的结果。

【样例输入 1】

```
1 2 4
1 2
```

【样例输出 1】

```
1 9 16
```

【样例解释 1】

图中共有 $(1,1),(2,1),(2,2)$ 这三条边，其中大小为 0 的好子集只有空集，其权值为 1，故所有大小为 0 的好子集的权值和只有 1；大小为 1 的好子集有 $(1,1),(1,2),(2,2)$，权值分别为 $4,1,4$，权值和为 9；大小为 2 的好子集只有有 $(1,1),(2,2)$，其权值为 16，故权值和为 16。

【样例输入 2】

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

【样例输出 2】

```
1 34 404 2076 4632 3936 864
```

【样例输入/输出 3】

见附件中的 ***story3.in/ans***，该样例满足测试点 2 的限制。

【样例输入/输出 4】

见附件中的 ***story4.in/ans***，该样例满足测试点 3 的限制。

【样例输入/输出 5】

见附件中的 ***story5.in/ans***，该样例满足测试点 7 的限制。

【样例输入/输出 5】

见附件中的 ***story6.in/ans***，该样例满足测试点 9 的限制。

【测试点约束】

对于所有测试点均满足：$1 \le n \le 2000, 0 \le y < 998244353, 0 \le a_i \le n$，序列 a 单调不降或单调不升。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	特殊性质	测试点分值
1	8	A	8
2	15	A	7
3	2000	AC	11
$4 \sim 5$	2000	A	12
6	15	B	6
$7 \sim 8$	2000	BD	10
$9 \sim 10$	2000	B	12

特殊性质 A: 保证序列单调不降。

特殊性质 B: 保证序列单调不升。

特殊性质 C: 保证 $y = 1$。

特殊性质 D: 令 $b_i = \sum_{j=1}^{n} [a_j \geq i]$，对于所有满足 $a_i \geq i$ 的 i，$a_i \geq b_i$。