

csp2023模拟赛

题目名称	牛奶香浓	商贸	投食	图
题目类型	传统型	传统型	传统型	传统型
可执行文件名	milky	trade	easiest	graph
输入文件名	milky.in	trade.in	easiest	graph.in
输出文件名	milky.out	trade.out	easiest.out	graph.out
每个测试点时限	2.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限	512 MiB	512 MiB	512 MiB	512 MiB
子任务/测试点数目	10	10	25	10
是否等分	是	是	是	是

提交源文件程序名

对于C++语言	milky.cpp	trade.cpp	easiest.cpp	graph.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。

牛奶香浓 (milky)

【题目描述】

FJ(作为一名Oier, 你应该知道他的全名) 决定带你去参观他最新研发的挤奶方式!

有 n 只奶牛, 每只上面有一个字母 $t_i \in \{m, i, l, k, y\}$, FJ 要想在 $[l, r]$ 中挤出他认为很 milky 的牛奶, 只需要:

- 选择 $l \leq a < b < c < d < e \leq r$, 满足这些编号的奶牛上面的字母从左到右连起来是 *milky*, 然后 FJ 会依次该他们挤奶, 最后大喊 "milky!" 即可挤出很 milky 的牛奶。

现在, FJ 每次给你 $[l, r]$, 问你有多少种方式挤出 milky 的牛奶! 答案对于 2^{32} 取模。

【输入格式】

第一行, 一个长度为 n 的字符串 T , 表示每只奶牛上面的字符。

第二行, 一个数 Q , 表示询问次数。

接下来 Q 行, 每一行两个数, 表示 l, r 。

【输出格式】

输出 Q 行, 对于每一个询问表示答案。

【输入输出样例1】

milky.in	milky.out
milkymilky	
5	1
1 5	1
1 6	0
5 9	0
4 4	6
1 10	

【数据规模与约定】

对于所有测试数据, 满足 $1 \leq n, Q \leq 10^6, 1 \leq l \leq r \leq n, t_i \in \{m, i, l, k, y\}$ 。

测试点编号	$N \leq$	$Q \leq$	特殊性质
1 ~ 3	1000	1000	
4 ~ 5	10^5	10^5	
6	10^6	10^5	
7	10^6	10^6	$l = 1$
8 ~ 10	10^6	10^6	

商贸 (trade)

【题目描述】

勇者小 A 通过寻宝提升自己的实力之后仍然无法战胜虫群，他终于意识到了一个人的力量终究时有极限的，因此，他决定开始招兵买马，然而，一个很现实的问题摆在他的面前——他没有钱。所幸他现在所在的国家是一个商贸非常发达的商业共和国，他可以从商赚钱。

共和国内有 N 座城市，编号为 $1, 2, 3, \dots, N$ 。由 M 条**单向**的道路连接。小 A 每一次到达城市 i 都可以获得 m_i 的收益。小 A 最开始位于城市 1，最后也必须回到编号为 1 的城市。

不过身为勇者的小 A 的时间是非常宝贵的，沿着一条道路从一座城市到达另一座城市需要花费一天的时间。为了量化小 A 花费时间贸易的代价，我们认为小 A 花费 T 时间贸易的成本为 $C \times T^2$ 。你需要计算出小 A 可能得到的最高收益。

【输入格式】

第一行输入三个正整数 N, M, C ，含义见题面。

输入的第二行包括 N 个整数，第 i 个整数即为 m_i 。为了避免争议，我们保证 $m_1 = 0$ 。

接下来的 M 行，每行包含两个用空格隔开的正整数 $a, b (a \neq b)$ ，表示一条从城市 a 到城市 b 的**单向通道**。

【输出格式】

输出一行，表示小 A 通过贸易可以获得的最大收益。

【输入输出样例1】

trade.in	trade.out
3 3 1 0 10 20 1 2 2 3 3 1	24

【数据规模与约定】

对于所有的测试数据，保证 $2 \leq N \leq 1000, 1 \leq M \leq 2000, 1 \leq C \leq 1000, 0 \leq m_i \leq 1000$ 。

存在 30% 的测试数据， $N \leq 200, M \leq 500$ 。

投食(easiest)

【题目描述】

1. 操作规则

对于一个操作序列,现在要按照如下规则取数:

1. 如果一个数是操作序列的第一个数,那么这个数可取;
2. 如果一个数比它的前一个数小,那么这个数可取。

最后可以将整个操作序列取完。

2. 询问内容

现在给出一个 $1 \sim n$ 的排列 A , 现在对 A 有两种询问:

1 $l_1 \ r_1 \ l_2 \ r_2$: 操作序列是 $[l_1, r_1]$, 不取区间 $[l_2, r_2]$ 中的数最多可以取多少个数。

2 $l_1 \ r_1 \ l_2 \ r_2$: 操作序列是 $[l_1, r_1]$, 在区间 $[l_2, r_2]$ 中有多少点对 $(u, v) (u < v)$ 使得存在一种方案 u, v 被连续取出。

注意,这里的区间是位置的区间,不是值的区间。

请对于每个询问输出它们的答案。

【输入格式】

第一行两个数 n, m , 接下来一行 n 个数描述排列 A 。

接下来 m 行, 每行 5 个数描述询问。

【输出格式】

第 i 行一个数表示第 i 个询问的答案。

【输入输出样例1】

easiest.in	easiest.out
7 3	
2 7 3 4 1 6 5	2
1 2 6 4 6	4
1 1 7 4 5	9
2 1 7 3 7	

【输入输出样例 1 说明】

询问 2 1 7 3 7。

按照取数规则, 把 2 7 3 4 1 6 5 取完, 有如下方法: (下述排列为取数顺序)

2 7 3 1 4 5 6 其中 $(1, 3), (4, 5), (5, 6), (1, 4)$ 可以相邻取出。

2 7 3 4 1 5 6 其中 $(3, 4), (1, 5)$ 可以相邻取出。

2 7 3 1 4 6 5 其中 (4, 6) 可以相邻取出。

2 7 3 5 4 1 6 其中 (3, 5), (1, 6) 可以相邻取出。

所以答案是 9。

【数据规模与约定】

对于所有测试点满足 $1 \leq l_1 \leq l_2 \leq r_2 \leq r_1 \leq n$.

数据点	数据范围	特殊条件
1 ~ 2	$n, m \leq 10$	无
3 ~ 5	$n, m \leq 300$	只有操作 1
6 ~ 9	$n, m \leq 300$	无
10 ~ 12	$n, m \leq 3000$	只有操作 1
13 ~ 16	$n, m \leq 3000$	无
17 ~ 20	$n, m \leq 100000$	只有操作 1
21 ~ 25	$n, m \leq 100000$	无

图 (graph)

【题目描述】

小 M 有一张 n 个点的无向图。图上的节点 i 有一个二元组 (x_i, y_i) 作为点权。两点 $i, j (i \neq j)$ 之间有无向边当且仅当两个点的点权存在偏序关系, 即: $x_i \leq x_j$ 且 $y_i \leq y_j$, 或 $x_j \leq x_i$ 且 $y_j \leq y_i$ 。

求这张图的联通块个数。

【输入格式】

第一行一个正整数 n 。

接下来 n 行, 每行两个整数 x_i, y_i 。

【输出格式】

一行一个整数表示答案。

【输入输出样例1】

graph.in	graph
6 1 5 2 4 2 2 4 2 5 1 5 3	3

【数据规模与约定】

对于 40% 的数据, $n \leq 1000$ 。

对于 100% 的数据, $1 \leq n \leq 10^5, -10^9 \leq x_i, y_i \leq 10^9$ 。