
重庆市育才中学2022年 寒假联合模拟

Round 3

Feb 6th, 2022

源文件名均为小写,输入输出文件后缀为.in/.out

题目概述

中文题目名称	商贸	部队集结	虫群扩张	推演
英文题目名称	trade	gather	proliferate	calc
输入文件名	trade.in	gather.in	proliferate.in	calc.in
输出文件名	trade.out	gather.out	proliferate.out	calc.out
单个测试点时限	1s	1s	1s	1s
程序空间限制	512 MB	512 MB	512 MB	512 MB

商贸 (trade.cpp/in/out)

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，他可以从中跑商赚钱。

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。

Input

第一行输入三个正整数 N, M, C ($2 \leq N \leq 1000, 1 \leq M \leq 2000, 1 \leq C \leq 1000$)，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 ($0 \leq m_i \leq 1000$)。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。

Output

输出一行，表示小A通过贸易可以获得的最大收益。

Scoring

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

Example

input	output
3 3 1	24
0 10 20	
1 2	
2 3	
3 1	

样例解释：最优旅行方案为 $1, 2, 3, 1, 2, 3, 1$ 。利润为24。

部队集结 (`gather.cpp/in/out`)

小A需要集结他的部队。通过使用钞能力，小A已经招募到了 N 名常年游走于刀剑之上的老练雇佣兵，现在，他们拍成一列，编号从1到 N ，正在接受小A的检阅。

小A也需要将这 N 名雇佣兵分成小队行动，但是，领队的选择成为了打问题。具体来说，小A只能选择一个连续的编号区间 $[l, r]$ 作为小队行动，所有编号 $l \leq i \leq r$ 的雇佣兵 i 会加入小队，雇佣兵 l, r 会作为领队。然而，不同的雇佣兵有自己的技术倾向，具体来说，我们可以使用数字 b_i 来描述编号为 i 的雇佣兵的技术倾向，如果队长的技术倾向与队内的其他成员(包括另一位队长)相同，那么这个队伍就无法和睦相处。

你的任务是帮助小A计算出有多少个区间可以划分出一个和睦相处的小队。

Input

输入的第一行包括一个整数 $N(1 \leq N \leq 2 \times 10^5)$ 。含义见题面。

第二行包含 N 个整数 $b_1, b_2, b_3 \dots b_N(1 \leq b_i \leq N)$ 。含义见题面。

Output

输出一行一个正整数表示可以划分出和睦相处小队的区间的数目。

Scoring

数据有梯度。

存在15%的测试数据， $N \leq 10$ 。

另有25%的测试数据， $N \leq 5000$ 。

Example

input

7
1 2 3 4 3 2 5

output

13

样例解释：合法的区间包括：

$[1,2], [1,3], [1,4], [1,7], [2,3], [2,4], [3,4], [4,5], [4,6], [4,7], [5,6], [5,7], [6,7]$ 。

虫群扩张(proliferate.cpp/in/out)

经过一番战斗之后，虫群终于被小A击溃，虫群女王也落入到了小A设计的陷阱之中。

小A设计的陷阱可以用一个 $N \times N$ 的方阵表示，其中，每一个方格可能是空格或者炸弹，并且所有边界上的方格都被小A布置了炸弹。某一些没有被布置炸弹的方格可能回使虫群女王出现的起始位置。

最开始，虫群女王位于某一个可能的起始位置上，在这之后的每一个单位时间内，女王和她的虫群会向着相同的方向移动一格或停留在原地（移动方向为网格内的上下左右四个方向中的一个）。每经过 D 个单位时间，虫群会进行一个增殖，位于 (x, y) 的虫子会在方格 $(x + 1, y), (x - 1, y), (x, y - 1), (x, y + 1)$ 各产生一只新的虫子。每一个格子内可能出现多只虫子。

如果移动或者增殖使得任何一只虫子触碰到了炸弹，那么炸弹就会被引爆，由于连锁爆炸，整个网格图都会被炸毁。身为慎重勇者的小A具有短距离传送的能力，因此他决定在炸弹激发之前前往陷阱内以防止一些意外情况地发生。为了让小A地行动更加有效率，你需要帮助小A求出有多少个格子内可能出现虫群。

Input

第一行包含两个用空格隔开的正整数 N 和 D ($3 \leq N \leq 1000, 1 \leq D \leq 10^9$)，含义如题面所示。

接下来的 N 行，每行一个长度为 N 的字符串，表示网格图的一行，每一个字符为'.','#','S'中的一个，'.'和'S'均为空地，但是'S'是可能的虫群女王位于的起始位置。'#'则是被安放了炸弹的位置，第一行，第一列，最后一行，最后一列的所有字符均为'#'。

Output

输出一个整数，表示可能在某一个时刻出现虫群的方格的数量。

Scoring

存在10%的数据， $D = 10^9$ 。

另有15%的数据， $D = 1$ 。

另有20%的测试数据满足 $N \leq 100$ 。

Example

input 1 10 1 ##### #.....# #S.....# #.....# ##### #S....S..# #####	output 1 15
---	-----------------------

```
#####
#####
#####
```

input 2

```
10 2
#####
#.#.....#
#.#.....#
#S.....#
#.#.....#
#.#.....#
#####
#####
#####
#####
```

input 3

```
10 2
#####
#.S#.....#
#..#.....#
#S.....#
#..#.....#
#..#.....#
#####
#####
#####
#####
```

output 2

```
28
```

output 3

```
10
```

样例1解释:在下面的图中，我们用'x'表示可能出现虫群的位置：

```
#####
#xxx....#
#xxxx....#
#xxx....#
#####
#xx..xxx.#
#####
#####
#####
```

下方的情况比较显然.上方所有与起点距离小于等于2的'x'，都可以直接走到，剩下的三个格子可以在起点右侧两个的位置复制到达。

推演 (calc.cpp/in/out)

尽管虫群已经被小A彻底毁灭了，女王手中进入魔王城的四个钥匙碎片之一仍然没有被小A获得，这个钥匙碎片被藏在了虫巢内部一个隐蔽的传送门内的空间里。

这篇神奇的空间并不是一篇平坦的空间，这篇空间可以被看成是一个巨大的二维网格。对于其中的一个坐标 (x, y) ，只有在满足如下条件时才是一块可以到达的悬浮石块：

对于所有的整数 $k \geq 0$ ， $\lfloor \frac{x}{3^k} \rfloor$ 和 $\lfloor \frac{y}{3^k} \rfloor$ 对3取模得到的数字奇偶性相同。例如： $(1, 7)$ 就是一个满足条件的坐标，因为当 $k = 0$ 时两个数字对3取模得到的结果都是1， $k = 1$ 时两个数字对3取模得到的结果为0和2。奇偶性相同。而 $k \geq 2$ 时，得到的数都是0。

由于自身技能特性的原因，小A需要你计算出某一个特定区域内有多少个格子上是有悬浮石块的。具体来说，他会给出 Q 个询问，每一个询问包括三个整数 x_i, y_i, d_i 。对于每一个询问，小A想知道有多少悬浮石块位于 (x_i, y_i) 到 $(x_i + d, y_i + d)$ 的对角线方格上(包括两个端点)。

Input

输入的第一行包含一个正整数 $Q (1 \leq Q \leq 10^4)$ 表示询问的数量。

接下来的 Q 行，每行包含三个整数 d_i, x_i 和 $y_i (0 \leq x_i, y_i, d_i \leq 10^{18})$ 表示一个询问。

Output

输出 Q 行，对于每一个询问输出一行一个整数表示询问的答案。

Scoring

存在10%的测试数据，满足 $d_i \leq 100$ 。

另有40%的测试数据，对于所有的测试数据，满足 $x_i + d = 3^{30} - 1, y_i = 0$

Examples

input 1	output 1
8	11
10 0 0	0
10 0 1	4
9 0 2	3
8 0 2	1
0 1 7	2
1 1 7	2
2 1 7	10000000000000000001
10000000000000000000 10000000000000000000	
10000000000000000000	