

# NOIP 模拟赛

HSEFZ 2024.10.21

题目名称	不连续子序列	造题	机器人	岛屿
题目类型	传统型	传统型	传统型	传统型
目录	sub	provide	robot	island
可执行文件名	sub	provide	robot	island
输入文件名	sub.in	provide.in	robot.in	island.in
输出文件名	sub.out	provide.out	robot.out	island.out
每个测试点时限	1 秒	1 秒	1 秒	2 秒
内存限制	512 MiB	512 MiB	512 MiB	1024 MiB
测试点数目	10	20	20	25
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	sub.cpp	provide.cpp	robot.cpp	island.cpp
-----------	---------	-------------	-----------	------------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static
-----------	------------------------

注意事项（请仔细阅读）

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
- 提交的程序代码文件直接放在选手目录下，无需开子文件夹。（建议子文件夹内外各放一份）
- 因违反以上三点而出现的错误或问题,申诉时一律不予受理。
- 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）。
- 选手提交的程序源文件必须不大于 100KB。
- 程序可使用的栈空间内存限制与题目的内存限制一致。
- 全国统一评测时采用的机器配置为: Intel(R) Core(TM) i5-9500 CPU @ 3.00GHz, 内存 16GB。上述时限以此配置为准。
- 评测在当前最新公布的 NOI Linux 下进行, 各语言的编译器版本以此为准。

# 不连续子序列 (sub)

## 题目描述

你有一个序列  $\{x_n\}$ ,  $x_i \in \{1, 2, 3\}$ 。

定义一个非空子序列是一个非空下标集合  $S \subseteq \{1, 2, \dots, n\}$ , 一个非空子序列是“非空不连续子序列”当且仅当它不是“非空连续子序列”, 即  $\exists \min S \leq i \leq \max S, i \notin S$ 。

定义一个非空子序列  $S$  是“好”的当且仅当  $\exists c \in \{1, 2, 3\}, \sum_{i \in S} [x_i = c] \geq \frac{|S|}{2}$ , 序列  $\{x_n\}$  是“完美”的当且仅当其每一个“非空不连续子序列”都是“好”的。注意, 长度  $\leq 2$  的序列必然是“完美”的, 因为它没有“非空不连续子序列”。

现在给定一个序列  $\{y_m\}$ ,  $y_i \in \{0, 1, 2, 3\}$  以及  $q$  次询问, 每次询问对于该序列的一个区间  $z = y_{[l,r]}$ , 在每种把  $z$  中每个 0 都替换成 1, 2, 3 中的一个的方案中, 有多少种得到的新序列  $x$  是“完美”的。

由于答案可能很大, 请输出其对 998 244 353 取模的结果。

## 输入格式

从文件 `sub.in` 读入数据。

第一行输入两个正整数, 代表  $m, q$ 。

第二行输入  $m$  个在 0, 1, 2, 3 范围内的数, 代表序列  $y$ 。

接下来  $q$  行, 每行两个正整数  $l, r$ , 代表一次询问  $z = y_{[l,r]}$ 。

## 输出格式

输出到文件 `sub.out` 中。

$q$  行, 每行一个非负整数, 代表每次询问答案对 998 244 353 取模的结果。

## 样例 1 输入

```
6 3
2 0 2 1 0 3
1 6
1 3
4 6
```

## 样例 1 输出

```
0
3
3
```

## 样例 1 解释

对于第一组询问，不管填入什么，“非空不连续子序列”  $\{1, 4, 6\}$  都不为“好”的，因此答案为 0。

## 样例 2

见选手目录下的 `sub/sub2.in` 与 `sub/sub2.ans`。

该组样例满足数据范围中的测试点 1 的限制。

## 样例 3

见选手目录下的 `sub/sub3.in` 与 `sub/sub3.ans`。

该组样例满足数据范围中的测试点 2 的限制。

## 样例 4

见选手目录下的 `sub/sub4.in` 与 `sub/sub4.ans`。

该组样例满足数据范围中的测试点 3, 4 的限制。

## 样例 5

见选手目录下的 `sub/sub5.in` 与 `sub/sub5.ans`。

该组样例满足数据范围中的测试点 5 的限制。

## 样例 6

见选手目录下的 `sub/sub6.in` 与 `sub/sub6.ans`。

该组样例满足数据范围中的测试点 6 ~ 10 的限制。

## 数据范围

对于所有测试数据，保证：

- $1 \leq m, q \leq 10^6$ ,
- $\forall 1 \leq i \leq m, y_i \in \{0, 1, 2, 3\}$ ,
- $1 \leq l \leq r \leq m$ 。

测试点	$m \leq$	$q \leq$	特殊性质
1	6	$10^6$	无
2	100	100	A
3, 4	100	100	B
5	100	100	无
6 ~ 10	$10^6$	$10^6$	无

特殊性质 A：保证  $y_i \in \{1, 2, 3\}$ 。

特殊性质 B：保证  $y_i \in \{0, 1, 2\}$ 。

# 造题 (provide)

## 题目背景

听说模拟赛分数高一点能让大家更开心，于是出现了这道水题。

## 题目描述

但是小 X 在题目准备好之后，意识到模拟赛分数太高了也不好，所以出题人小 X 想要删除一些题目以降低选手分数。

具体而言，认为一套模拟赛是一个  $R$  行  $C$  列的网格，其中第  $i$  行第  $j$  个格子为  $(i, j)$  ( $1 \leq i \leq R, 1 \leq j \leq C$ )。每个格子内各有一道题目，格子  $(i, j)$  内的题目分值为非负整数  $A_{i,j}$  ( $0 \leq A_{i,j} \leq 10^9$ )。为了降低选手得分，小 X 可以恰好选择  $R + C - 1$  个格子，并且删除这些格子内的题目，要求满足：

- 格子  $(1, C)$  和  $(R, 1)$  必须被选中。
- 若  $(R, 1)$  之外的某个格子  $(i, j)$  被选中，则  $(i + 1, j)$  和  $(i, j - 1)$  中有且仅有一个格子被选中（若其中仅有一个格子存在，则这个格子必须被选中）。

由于模拟赛时间有限，选手小 Y 也需要恰好选择  $R + C - 1$  个格子，并且完成这些格子内**未被删除**的题目，获得等于这些题目的分值之和的分数（即小 Y 既不会挂分也不会在其余题目得分），要求满足：

- 格子  $(1, 1)$  和  $(R, C)$  必须被选中。
- 若  $(R, C)$  之外的某个格子  $(i, j)$  被选中，则  $(i + 1, j)$  和  $(i, j + 1)$  中有且仅有一个格子被选中（若其中仅有一个格子存在，则这个格子必须被选中）。

如果将网格视为一个  $R \times C$  个结点的图，其中每个格子和结点一一对应，且结点  $(x_1, y_1)$  和结点  $(x_2, y_2)$  ( $1 \leq x_1, x_2 \leq R, 1 \leq y_1, y_2 \leq C$ ) 间有一条无向无权边当且仅当  $|x_1 - x_2| + |y_1 - y_2| = 1$ ，则小 X 选中的格子对应结点构成一条  $(1, C)$  和  $(R, 1)$  间的最短路径，小 Y 选中的格子对应结点构成一条  $(1, 1)$  和  $(R, C)$  间的最短路径。

注意模拟赛总是在题目准备完成之后，因此小 Y 在小 X 删除了如上所述的  $R + C - 1$  道题目后再开始打模拟赛。小 Y 希望最大化自己获得的分数，小 X 希望最小化小 Y 获得的分数，且两人都按最优策略操作。求最优策略下小 Y 最终的分数。

## 输入格式

从文件 `provide.in` 读入数据。

输入的第一行包含两个正整数  $R, C$ ，分别表示网格的行数和列数。

接下来  $R$  行，每行  $C$  个非负整数。其中第  $i$  行的第  $j$  个非负整数  $A_{i,j}$  表示网格第  $i$  行第  $j$  个格子内的题目分值。

## 输出格式

输出到文件 `provide.out` 中。

输出共一行一个非负整数，表示在最优策略下小 Y 最终获得的分数。

## 样例 1 输入

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

## 样例 1 输出

```
12
```

## 样例 1 解释

其中一种最优策略是：小 X 选中格子  $(1, 4), (1, 3), (1, 2), (2, 2), (2, 1), (3, 1)$ ，小 Y 选中格子  $(1, 1), (2, 1), (2, 2), (2, 3), (2, 4), (3, 4)$ 。小 Y 选择的格子中，格子  $(1, 1), (2, 3), (2, 4), (3, 4)$  内的题目未被删除，因此小 Y 获得的分数为  $2 + 2 + 5 + 3 = 12$ 。

## 样例 2 输入

```
2 2
2 2
2 1
```

## 样例 2 输出

```
1
```

## 样例 3

见选手目录下的 `provide/provide3.in` 与 `provide/provide3.ans`。

## 样例 4

见选手目录下 `provide/provide4.in` 与 `provide/provide4.ans`。

## 样例 5

见选手目录下 `provide/provide5.in` 与 `provide/provide5.ans`。

## 数据范围

对于所有测试数据，保证：

- $2 \leq R, C \leq 300$ ,
- $0 \leq A_{i,j} \leq 10^9$ 。

测试点编号	$R \leq$	$C \leq$	$A_{i,j} \in$
1 ~ 2	10	10	$[0, 10^9]$
3 ~ 5	20	20	$[0, 10^9]$
6 ~ 8	50	50	$[0, 10^9]$
9 ~ 12	100	100	$[0, 10^9]$
13 ~ 14	100	300	$[0, 10^9]$
15 ~ 16	300	300	$[1, 1]$
17 ~ 20	300	300	$[0, 10^9]$

## 提示

十年 OI 一场空，不开 long long 见祖宗。

# 机器人 (robot)

## 题目描述

在二维平面上有一个机器人，初始位置为  $(0, 0)$ ，面向  $x$  轴正方向（下文将该位置、方向称为**初始状态**）。

机器人每次可以执行一段操作序列。操作序列为一个的字符串，每个字符均为 `W`, `A`, `S`, `D` 之一，从左到右依次执行每个字符代表的指令。

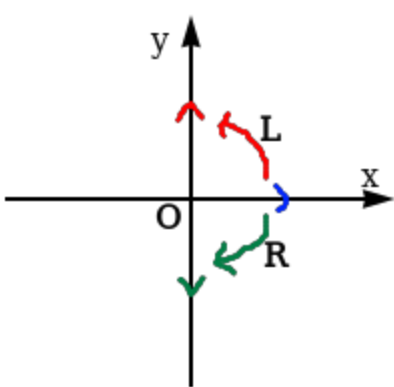
以下为每个字符所代表的指令：

- `W`：机器人从当前位置向前一步；
- `A`：向左转，即机器人逆时针旋转  $90^\circ$ ；
- `S`：机器人从当前位置向后一步；
- `D`：向右转，即机器人顺时针旋转  $90^\circ$ 。

例如：

- 机器人从初始状态，执行指令 `W` 之后：位于  $(1, 0)$ ，面向  $x$  轴正方向；
- 机器人从初始状态，执行指令 `A` 之后：位于  $(0, 0)$ ，面向  $y$  轴正方向；
- 机器人从初始状态，执行指令 `S` 之后：位于  $(-1, 0)$ ，面向  $x$  轴正方向；
- 机器人从初始状态，执行指令 `D` 之后：位于  $(0, 0)$ ，面向  $y$  轴**负**方向。

如图所示，其中蓝色、红色、绿色箭头分别表示机器人初始面向方向、执行指令 `A` 后面向方向、执行指令 `D` 后面向方向。



输入一个长度为  $N$  的字符串  $A$ ，保证  $A$  每个字符都是 `W`, `A`, `S`, `D` 之一，下标从 1 开始，即  $A$  的  $N$  个字符依次为  $A_1, A_2, \dots, A_N$ 。

$Q$  次操作，每次操作形如：

- `1 x ch`，其中  $x$  为  $[1, N]$  内的正整数， $ch$  为字符 `W`, `A`, `S`, `D` 之一，表示将  $A_x$  修改为  $ch$ 。



- `2 l r`，其中  $l, r$  为  $[1, N]$  内的正整数且  $l \leq r$ ，表示选择字符串  $A_l, A_{l+1}, \dots, A_r$  作为操作序列，使得机器人从初始状态开始执行，输出机器人最终处于的位置。

## 输入格式

从文件 `robot.in` 读入数据。

输入的第一行包含一个正整数  $N$ ，表示字符串长度。

第二行包含一个字符串  $A$ ，含义见上。

第三行包含一个正整数  $Q$ ，表示询问组数。

接下来  $Q$  行，每行若干整数或大写字母，形如 `1 x ch` 或 `2 l r`，表示依次操作。

## 输出格式

输出到文件 `robot.out` 中。

若干行，依次输出每个操作 `2` 的答案，即对于每个操作 `2`，输出一行两个整数  $x, y$ ，表示机器人在执行相应操作序列后位置为  $(x, y)$ 。

## 样例 1 输入

```
4
WASD
14
2 1 1
2 2 2
2 3 3
2 4 4
1 1 A
1 2 S
1 3 D
1 4 W
2 1 2
2 2 3
2 3 4
2 1 3
2 2 4
2 1 4
```

## 样例 1 输出

```
1 0
0 0
-1 0
0 0
0 -1
-1 0
0 -1
0 -1
-1 -1
1 -1
```

## 样例 2 输入

```
5
WWWWA
10
1 2 W
1 2 W
2 3 5
1 5 W
1 2 D
1 5 D
1 5 S
2 2 5
1 2 A
2 1 4
```

## 样例 2 输出

```
2 0
0 -1
1 2
```

## 样例 3

见选手目录下的 `robot/robot3.in` 与 `robot/robot3.ans`。

该组样例满足数据范围中的测试点 1 的限制。

## 样例 4

见选手目录下 `robot/robot4.in` 与 `robot/robot4.ans`。

该组样例满足数据范围中的测试点 2 的限制。

## 样例 5

见选手目录下 `robot/robot5.in` 与 `robot/robot5.ans`。

该组样例满足数据范围中的测试点 4 的限制。

## 样例 6

见选手目录下 `robot/robot6.in` 与 `robot/robot6.ans`。

该组样例满足数据范围中的测试点 5 ~ 6 的限制。

## 样例 7

见选手目录下 `robot/robot7.in` 与 `robot/robot7.ans`。

该组样例满足数据范围中的测试点 7 ~ 8 的限制。

## 样例 8

见选手目录下 `robot/robot8.in` 与 `robot/robot8.ans`。

该组样例满足数据范围中的测试点 9 的限制。

## 样例 9

见选手目录下 `robot/robot9.in` 与 `robot/robot9.ans`。

该组样例满足数据范围中的测试点 10 ~ 11 的限制。

## 样例 10

见选手目录下 `robot/robot10.in` 与 `robot/robot10.ans`。

该组样例满足数据范围中的测试点 19 ~ 20 的限制。

# 数据范围

对于所有测试数据，保证：

- $1 \leq N, Q \leq 5 \times 10^5$ ,
- $1 \leq l \leq r \leq N$ ,
- $1 \leq x \leq N$ ,
- $A_i, ch \in \{W, A, S, D\}$ 。

测试点编号	$N \leq$	$Q \leq$	$A_i, ch \in$	特殊性质
1	100	100	$\{W, A, S, D\}$	无
2	300	300	$\{W, A, S, D\}$	无
3	5000	1	$\{W, A, S, D\}$	A
4	5000	5000	$\{W, A, S, D\}$	A
5 ~ 6	5000	5000	$\{W, A, S, D\}$	无
7 ~ 8	$8 \times 10^4$	$8 \times 10^4$	$\{W, A, S, D\}$	A
9	$8 \times 10^4$	$8 \times 10^4$	$\{W, S\}$	无
10 ~ 11	$8 \times 10^4$	$8 \times 10^4$	$\{W, A, S\}$	无
12 ~ 13	$8 \times 10^4$	$8 \times 10^4$	$\{W, A, S, D\}$	无
14 ~ 15	$2 \times 10^5$	$2 \times 10^5$	$\{W, A, S, D\}$	A
16	$2 \times 10^5$	$2 \times 10^5$	$\{W, S\}$	无
17 ~ 18	$2 \times 10^5$	$2 \times 10^5$	$\{W, A, S\}$	无
19 ~ 20	$2 \times 10^5$	$2 \times 10^5$	$\{W, A, S, D\}$	无

特殊性质 A：无 **1** 操作。

# 岛屿 (island)

## 题目描述

给你一张  $N$  行  $M$  列的地图  $A$ ，记点  $(x, y)$  表示地图第  $x$  行第  $y$  列。每个点是海 `.`，岛屿 `#` 或者火山 `v`。并且保证：

- 岛屿和非岛屿均分别可以形成恰好一个四连通块；
- 第 1,  $N$  行以及第 1,  $M$  列都没有岛屿点；
- 至少有一个岛屿点与一个火山点。

定义一条合法的**环状**路径  $(x_1, y_1), \dots, (x_k, y_k)$  为：

- $\forall i \in [1, k], x_i \in [1, n], y_i \in [1, m], A_{x_i, y_i} \neq \#$ ;
- $x_1 = x_k, y_1 = y_k$ ;
- $\forall i \in [1, k), |x_i - x_{i+1}| + |y_i - y_{i+1}| = 1$ ;
- 若删去所有路径上点，则不存在一个 `#` 点与边界上的点在一个八连通块中。

定义这条路径的权值为：

$$\min_{A_{p,q}=v, i \in [1, k]} |p - x_i| + |q - y_i|$$

现有  $Q$  次询问，每次询问给定  $(x, y)$ ，求权值最大的包含点  $(x, y)$  的路径。输出最大权值。

## 输入格式

从文件 `island.in` 读入数据。

输入的第一行包含三个正整数  $N, M, Q$ ，表示地图的行数、列数和询问次数。

接下来  $N$  行，每行一个长度为  $M$  且仅包含字符 `.`, `#`, `v` 的字符串，依次表示地图  $A$  的第  $1, 2, \dots, N$  行。

接下来  $Q$  行，每行两个正整数  $x, y$ ，表示一组询问。

## 输出格式

输出到文件 `island.out` 中。

$Q$  行，依次输出每个询问的答案，即对于每个询问，输出一行一个非负整数，表示最大权值。

# 样例 1 输入

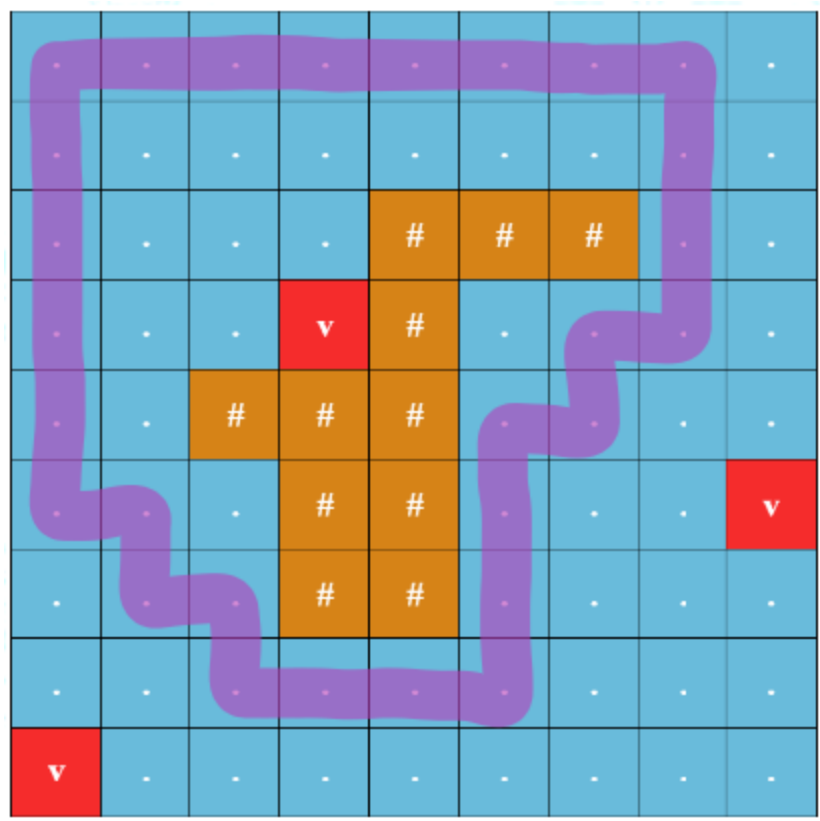
```
9 9 3
.....
.....
....###..
...v#....
..###....
...##...v
...##....
.....
v.....
1 1
9 1
5 7
```

# 样例 1 输出

```
3
0
3
```

# 样例 1 解释

对于第 1 组和第 3 组询问，一种方案如图所示。



样例 2 输入

```
3 3 5
..v
.#.
...
1 2
1 3
2 3
2 1
3 2
```

样例 2 输出

```
0
0
0
0
0
```

### 样例 3 输入

```
14 13 5
.....
.....
.....
...VVVVVVV...
...V.....V...
...V.###.V...
...V.#.#.V...
...V..V..V...
...V..V..V...
....V...V....
.....VVV.....
.....
.....
.....
1 1
7 7
5 6
4 10
13 6
```

### 样例 3 输出

```
3
0
1
0
2
```



## 样例 4 输入

```
10 11 4
.....
..#####..
..#..#..#..
..#.....#..
..#..v..#..
..#..###.#..
..#..#..#..
..#...#.#..
..#####.#..
.....
7 6
3 7
6 8
1 1
```

## 样例 4 输出

```
1
2
3
4
```

## 样例 5

见选手目录下的 `island/island5.in` 与 `island/island5.ans`。

## 数据范围

对于所有测试数据，保证：

- $1 \leq N, M \leq 500$ ;
- $1 \leq Q \leq 3 \times 10^5$ ;
- $1 \leq x \leq N$ ;
- $1 \leq y \leq M$ ;
- 岛屿和非岛屿均分别可以形成恰好一个四连通块；
- 第 1,  $N$  行以及第 1,  $M$  都没有岛屿点；
- 至少有一个岛屿点与一个火山点。

测试点编号	$N, M \leq$	$Q \leq$
1	5	5
2 ~ 3	5	15
4	10	5
5 ~ 6	10	75
7	20	5
8 ~ 9	20	5
10	50	5
11 ~ 12	50	2500
13	100	5
14 ~ 15	100	10000
16	200	5
17 ~ 18	200	40000
19	300	5
20 ~ 21	300	90000
22	500	5
23 ~ 25	500	$3 \times 10^5$