

NOIP 模拟赛

| 题目名称 | 星空列车与白的旅行 | 极限脱出 | 布莱克寿司 | 俄罗斯方块 |
|-----------|----------------|------------|----------------|------------|
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |
| 可执行文件名 | hoshitetsu | escape | blacksouls | nanami |
| 输入文件名 | hoshitetsu.in | escape.in | blacksouls.in | nanami.in |
| 输出文件名 | hoshitetsu.out | escape.out | blacksouls.out | nanami.out |
| 每个测试点时限 | 1 秒 | 1 秒 | 1 秒 | 1 秒 |
| 内存限制 | 512 MB | 512 MB | 512 MB | 512 MB |
| 子任务/测试点数目 | 5 | 10 | 6 | 8 |
| 是否等分 | 是 | 否 | 否 | 否 |

提交源文件程序名

| | | | | |
|---------|----------------|------------|----------------|------------|
| 对于C++语言 | hoshitetsu.cpp | escape.cpp | blacksouls.cpp | nanami.cpp |
|---------|----------------|------------|----------------|------------|

编译选项

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

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/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。
- 选手提交的程序代码文件请**直接放在个人目录下，不需要建立子文件夹**。
- 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 50KB。
- 程序可使用的栈空间内存限制于题目的内存限制一致。
- 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。
- 题目较简单，AK 的选手不要大声喧哗。
- 请各位选手注意 IOI 2025 的相关事宜，不要携带电子设备进入考场。

星空列车与白的旅行 (hoshitetsu)

题目背景

这个世界就像是一个巨大的宝石箱，里面塞满了熠熠生辉的宝石。

题目描述

在一个有 n 个点， m 条无向边的图上，诺瓦想乘坐列车从 1 到点 n 。

一共有 s 条列车线路，第 i 条线路会依次经过 c_i 个点，第 j 站为 $a_{i,j}$ ，列车从一点到另一点所花的时间为两点之间的边权（保证这两点互不相同且一定有且只有一条边）。

每条线路都会有一个发车时间 x 和一个发车间隔 y ，第一班次线路的列车会在 x 时刻出发，在此之后每两班列车的发车间隔为 y 。

诺瓦可以在列车到达任意点的时候下车，也可以在列车到达她所在的点的时候上车，换乘的时间可以忽略不计。

由于诺瓦身体不好，所以她希望换乘次数（也就是上车次数 -1 ）不超过 k 。

她想知道她在 t 时刻出发，最早什么时候到点 n 。

输入格式

第一行五个整数 n, m, s, k, t 。

接下来 m 行，每行三个数 u, v, w 表示图上有一条点 u 到点 v ，边权为 w 的边。

接下来 $2s$ 行，第 $2i - 1$ 行三个数分别表示 c_i, x, y 。接下来一行 c_i 个数第 j 个表示 $a_{i,j}$ 。

输出格式

一行一个数，表示最早到点 n 的时间，如果无法在 k 次换乘内到达点 n ，输出 -1 。

样例 #1

样例输入 #1

```
4 4 2 1 1
1 2 2
2 3 4
1 3 3
4 3 2
4 0 10
1 2 3 4
3 2 7
1 3 2
```

样例输出 #1

```
8
```

提示

【数据范围】

对于所有测试数据，保证 $1 \leq n \leq 10000, 1 \leq m \leq 50000, 1 \leq s \leq 25000, 0 \leq k \leq 100, 0 \leq t, x \leq 10^9, 1 \leq y, w \leq 10^9, 2 \leq c \leq n, \sum c \leq 50000, 1 \leq a, u, v \leq n$ 。

| 子任务编号 | 限制 | 分数 | 子任务依赖 |
|-------|----------------------------------|----|-------|
| 1 | $k = n$ | 20 | 无 |
| 2 | $\forall i, a_{i,j} < a_{i,j+1}$ | 20 | 无 |
| 3 | $c = 2$ | 20 | 无 |
| 4 | $t = 0, x = 0, y = 1$ | 20 | 无 |
| 5 | 无特殊限制 | 20 | 1 ~ 4 |

极限脱出 (escape)

题目背景

akane 又被零抓去参加九人游戏了！

题目描述

零给了 akane 一个长为 n 的序列 a 和一个整数 m 。

定义：

- $f(l, r) = \sum_{i=l}^r a_i$
- $a[l \dots r]$ 为一个长度为 $r - l + 1$ 的新序列，其中第 i 项为 a_{l+i-1}
- $g(b, x)$ 表示将序列 b 从大到小排序后的第 x 项
- $h(l, r, k) = \sum_{i=1}^k g(a[l \dots r], i)$
- $F(l, r, k) = f(l, r) - h(l, r, k)$

akane 需要找到两个数 l, r 满足 $l + m - 1 \leq r$ 且 $1 \leq l \leq r \leq n$ ，并且最大化 $F(l, r, m)$ 。

零希望你告诉他最大的 $F(l, r, m)$ 是多少以验证 akane 的答案是不是正确的。

输入格式

第一行两个整数 n, m 。

接下来一行，一共 n 个数，表示序列 a 。

输出格式

一行一个数，表示最大的 $F(l, r, m)$ 。

样例 #1

样例输入 #1

```
5 0
2 -4 5 -1 7
```

样例输出 #1

```
11
```

样例 #2

样例输入 #2

```
5 1
2 -4 5 -1 7
```

样例输出 #2

4

样例 #3

样例输入 #3

5 2
2 -4 5 -1 7

样例输出 #3

0

提示

【样例解释 #1】

选择 $l = 3, r = 5$, $f(l, r) = 11, h(l, r, m) = 0, F(l, r, m) = 11$ 。

【样例解释 #2】

选择 $l = 3, r = 5$, $f(l, r) = 11, h(l, r, m) = 7, F(l, r, m) = 4$ 。

【样例解释 #3】

选择 $l = 1, r = 2$, $f(l, r) = -2, h(l, r, m) = -2, F(l, r, m) = 0$ 。

【数据范围】

对于所有测试数据，保证 $1 \leq n \leq 2 \times 10^5$, $0 \leq m \leq \min(100, n)$, $-10^9 \leq a \leq 10^9$ 。

| 子任务编号 | 限制 | 分数 | 子任务依赖 |
|-------|------------------------|----|----------|
| 1 | $n \leq 200$ | 5 | 无 |
| 2 | $n \leq 1000$ | 5 | 1 |
| 3 | $n \leq 6000$ | 10 | 1, 2 |
| 4 | $m = 0$ | 5 | 无 |
| 5 | $m = 1$ | 5 | 4 |
| 6 | $m = 2$ | 10 | 4, 5 |
| 7 | $n \leq 80000$ | 10 | 1 ~ 3 |
| 8 | 保证 a 中负数个数不大于 1000 个 | 10 | 1, 2 |
| 9 | 保证 a 中负数个数不大于 6000 个 | 10 | 1 ~ 3, 8 |
| 10 | 无特殊限制 | 30 | 1 ~ 9 |

布莱克寿司 (blacksouls)

题目背景

题目描述

爱丽丝知道格林喜欢做数学题，于是给格林出了一道数学题。

给定 a, n 求 $\lfloor (\lceil \sqrt{a} \rceil + \sqrt{a})^n \rfloor$

由于答案可能过大，所以只需要输出对 998244353 取模的结果即可。

因为格林做题速度太快了，所以爱丽丝一共会询问 q 次。

输入格式

第一行一个正整数 q 。

接下来 q 行，每行两个整数 a, n 。

输出格式

一共 q 行，每行一个整数表示对应的询问的答案。

样例 #1

样例输入 #1

```
7
1 1
2 2
7 4
99 257
101300 721
18520504 19341116
1324240482 3996390528
```

样例输出 #1

```
2
11
1015
889346623
181741070
439797692
497944115
```

提示

【数据范围】

对于所有测试数据，保证 $1 \leq q \leq 10^5$ ， $1 \leq a, n \leq 10^{18}$ 。

| 子任务编号 | 限制 | 分数 | 子任务依赖 |
|-------|----------------------|----|-------|
| 1 | a 是完全平方数 | 10 | 无 |
| 2 | $n \leq 10$ | 10 | 无 |
| 3 | $n = 2^k$ | 20 | 无 |
| 4 | $a = 2$ | 20 | 无 |
| 5 | $q = 1, n \leq 10^6$ | 20 | 无 |
| 6 | 无特殊限制 | 20 | 1 ~ 5 |

俄罗斯方块 (nanami)

题目背景

Nanami 玩腻了俄罗斯方块，她决定去玩玩蔬菜斗恶兔。

题目描述

这个游戏的目标是要在一个 $n \times m$ 的迷宫中**获得**尽可能多的金币。

这个迷宫有以下几种元素：

- .：空地。
- #：墙壁。
- o：金币。
- X：地雷。
- S：起点。

游戏开始时，Nanami 会随机出现在一个起点处，她可以看到与自己八连通的格子是什么。特殊的，地雷和起点都**看起来**和空地一样。

每次行动，Nanami 可以选择进行移动操作或者离开迷宫。

如果选择离开迷宫则游戏立刻结束，Nanami **获得**她收集到的所有金币。

如果选择移动行动则选择一个四联通的非墙壁格子，然后进入这个格子。

如果是金币则她可以**收集**此金币（收集之后金币会消失，也就是这个格子会变为空地），如果是地雷则金币清空且立刻结束游戏。

由于 Nanami 是超高校级的游戏玩家，所以她会连接大数据获得迷宫的地图，她想她在最坏情况下可以**获得**多少金币。

输入格式

第一行两个正整数 n, m 。

接下来 n 行，每行 m 个字符，表示迷宫的构造。

输出格式

输出一个整数，表示最坏情况下可以获得的金币个数。

样例 #1

样例输入 #1

```
3 7
#####
#Soooo#
#####
```


样例输出 #1

4

样例 #2

样例输入 #2

```
3 8
#####
#SoXooS#
#####
```

样例输出 #2

1

样例 #3

样例输入 #3

```
7 18
#####
#.....#
#..o...SX.....o.#
#..o...X..X.....o.#
#..o....XS.....o.#
#.....#
#####
```

样例输出 #3

0

样例 #4

样例输入 #4

```
7 18
#####
#...#.....#
#..o...SX.....o.#
#..o...X..X.....o.#
#..o....XS.....o.#
#.....#.....#
#####
```

样例输出 #4

6

样例 #5

样例输入 #5

```
7 18
#####
#.....X..S....oo#
#####
#..O..S.X.....O.#
#####X#####
#o.....S...X.....#
#####
```

样例输出 #5

1

提示

【样例解释 #1】

由于 S 只有一个，因此起点固定，可以获得所有金币。

【样例解释 #2】

分别以左侧和右侧 S 为起点时的初始视野（@ 为起点）：

```
###   ###
#@o   o@#
###   ###
```

以左侧的 S 为起点时可获得 1 枚金币，以右侧的为起点时可获得 2 枚。因此在最坏情况下最多可获得 $\min(1, 2) = 1$ 枚金币。

【样例解释 #3】

无论起点是哪一个 S，初始视野均为：

```
...
.@.
...
```

由于不知道当前位置具体在地图上是哪里，因此最坏情况是踏入起点旁边的 x（视野中为 .）。这种情况下不能获得任何金币，答案为 0。

【样例解释 #4】

分别以左上和右下的 S 为起点时的初始视野：

```

#..   ...
.@.   .@.
...   ..#

```

由于初始视野不同，因此可以通过 # 位置的不同判断在地图上的具体位置。因此可以获得所有金币，答案为 6。

【样例解释 #5】

不妨先向左走 2 步。如果能够看到 o，那么可以判断出位于第四行，同时捡起左侧的金币。

否则仍无法判断位于第二行还是第六行。因此可以在向左 2 步的基础上往右 4 步（即走到起点右侧 2 步的位置）。如果视野右上是 .（在地图中为 #），那么可以判断位于第六行。这时可以安全地一直向左获得第二列的金币。如果视野不是 .，那么只需一直向右即可获得第十六列和第十七列的金币。

按这种方案得到的答案是最优的，为 $\min\{1, 1, 2\} = 1$ 。

不难发现，如果直接向右，那么可能直接进入 x——这种方案得到的答案为 0。

【数据范围】

保证迷宫的四周都是墙壁。

s 表示迷宫的起点数量。

对于所有测试数据，保证 $1 \leq n, m \leq 400, 1 \leq s \leq 60$ 。

| 子任务编号 | 限制 | 分数 | 子任务依赖 |
|-------|-----------------------------------|----|-------|
| 1 | $s = 1$, 没有地雷, 除了迷宫的四周没有墙壁 | 5 | 无 |
| 2 | $s = 1$ | 5 | 1 |
| 3 | $s = 2$ | 10 | 1, 2 |
| 4 | $n = 3$ | 10 | 无 |
| 5 | $1 \leq n, m \leq 250, s \leq 12$ | 20 | 无 |
| 6 | $1 \leq n, m \leq 400, s \leq 20$ | 15 | 5 |
| 7 | 只有一枚金币 | 10 | 无 |
| 8 | 无特殊限制 | 25 | 1 ~ 7 |