Задача 1. В кинозале

Будем сохранять занятость мест в кинозале в массиве A[1..*N*][1..*M*]. Общее количество вставаний подсчитываем в ALL.

Проверяем каждого входящего зрителя. Пусть у него номер ряда R, номер места P.

1. Если R равно 1 или P равно 1 или M, то отмечаем, что место A[R][P] занято, и переходим к следующему зрителю.

2. Подсчитываем с какой стороны до места P меньше встающих людей. Заметим, человек с места *i* встает, если места A[R][*i* – 1], A[R][*i*] и A[R][*i* + 1] уже заняты.
   а) Считаем количество встающих на местах со 2-го до P-1.
   б) Считаем количество встающих на местах с P+1 до M-1.

   Добавляем к ALL наименьшее из а) и б).
3. Отмечаем, что место A[R][P] занято.

Задача 2. Игра

Сохраним игровое поле в массиве A[1..*N*][1..*M*].

Общие замечания.

Рассмотрим некоторую текущую клетку *C* игрового поля. Пусть Игрок 1 – тот, чья очередь ходить из *C*, Игрок 2 – его соперник .

Если из *C* Игрок 1 не может сделать ход, то он проигрывает на *C*.

A

2	0	1	2
1	1	2	2
1	1	1	1

Проигрыш Игрока 1 в клетке A[3][4]

Если Игрок 1 своим ходом из *C* может встать на клетку, на которой выигрывает Игрок 2 (т.е. это Игрок 1 для *C*), то Игрок 1 выигрывает на *C*. Иначе Игрок 1 проигрывает на *C*, а Игрок 2 выигрывает.

A

2	0	1	2
1	1	2	2
1	1	1	1

Выигрыш Игрока 1 в клетке A[1][4]

В массиве B[1..*N*][1..*M*] отмечаем, какой игрок выигрывает в каждой клетке.

1. Начинаем заполнять массив B с клетки B[*N*][*M*], B[*N*][*M*]:=2
2. Просматриваем ряды снизу вверх, каждый ряд справа налево. Пытаемся сделать ход из текущей клетки A[*i*][*j*] и проверяем, кто выигрывает на новых клетках.
3. Если по окончании заполнения B[1][1] = 1, то в игре выигрывает Алиса, иначе выигрывает Боб.

A

2	0	1	2
1	1	2	2
1	1	1	1

B

?	0	0	0
0	0	0	0
0	0	0	2

A

2	0	1	2
1	1	2	2
1	1	1	1

B

?	0	0	0
0	0	0	0
0	0	1	2

A

2	0	1	2
1	1	2	2
1	1	1	1

B

?	0	1	1
2	1	2	2
1	2	1	2

Для получения возможной клетки окончания игры моделируем ход игры со стартовой клетки A[1][1], *R*:=1, *C*:=1.

1. Если A[*R*][*C*]=0 или сделать ход в пределах игрового поля нельзя, то финишная клетка найдена.
2. Если B[*R*][*C*] =2, то переходим на любую из допустимых клеток.
3. Если B[*R*][*C*] =1, то из двух возможных клеток выбираем клетку со значением 2.

Задача 3. Телепорты

I. Без использования телепортов

Представим этажи – вершинами графа, лестницы – ориентированными дугами, время на перемещение – вес дуги.

Кратчайшие пути от вершины 1 до вершины N и от вершины N до вершины 1 находятся с помощью алгоритма Дейкстры.

II. Учёт использованных телепортов

Отмечаем пользование телепортом в битовой маске $0 \le j < 2^K$.

телепорт №: **3** **2** 1 (2 и 3 использовались)

маска ***j* = 6**: **1** **1** 0

Строим новый граф, в котором

вершина = (номер этажа, битовая маска),

дуги – лестницы и телепорты

вес дуги – время на перемещение

В новом графе

1. находим кратчайшие пути от вершины (1, 0) до вершин (N, j), $0 \le j < 2^K$.
2. находим кратчайшие пути от вершины (N, 0) до вершин $(1, j)$, $0 \le j < 2^K$.
3. комбинируем ответы из пп. 1. и 2., чтобы использованные телепорты не пересекались.

Задача 4. Толя и картина

Сохраним расположение шурупов в массиве $A[1..N][1..M]$:

$$A[i][j] := \begin{cases} 1, & \text{если в точке } (i,j) \text{ есть шуруп} \\ 0, & \text{иначе} \end{cases}$$

Сохраним описание картины в массиве $P[1..R][1..C]$:

$$P[i][j] := \begin{cases} 1, & \text{если в точке } (i,j) \text{ есть символ ‘\#’} \\ 0, & \text{иначе} \end{cases}$$

Способ 1. Подставляем картину в каждую возможную точку и считаем количество покрытых шурупов.

Левую верхнюю точку картины можно подставить в точку (k,t),

если $k + R - 1 \le N$ и $t + C - 1 \le M$

Точка (i,j) картины покрывает шуруп, если

$P[i][j] = 1$ и $A[k+i][t+j] = 1$

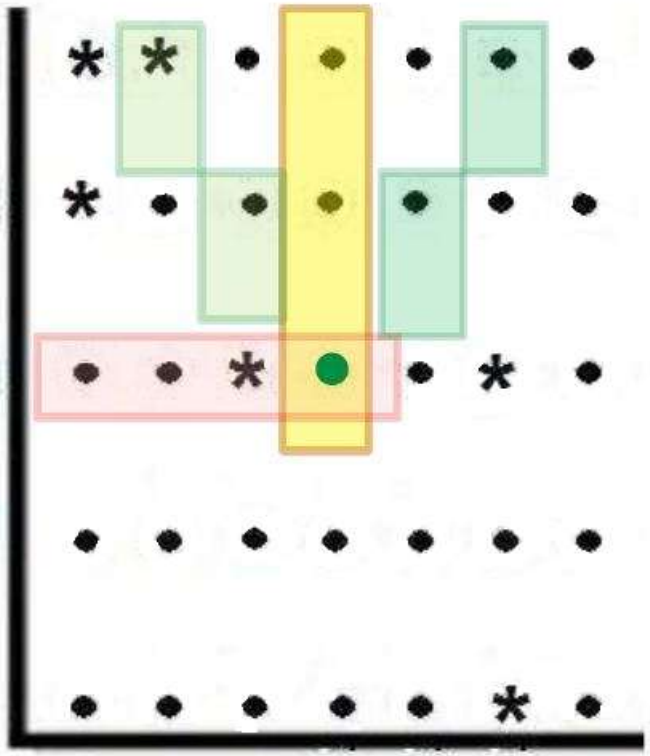
Временная сложность *способа 1* = $O(NMRC)$

Способ 2.

1. Проведем дополнительный подсчет шурупов с помощью частичных сумм.

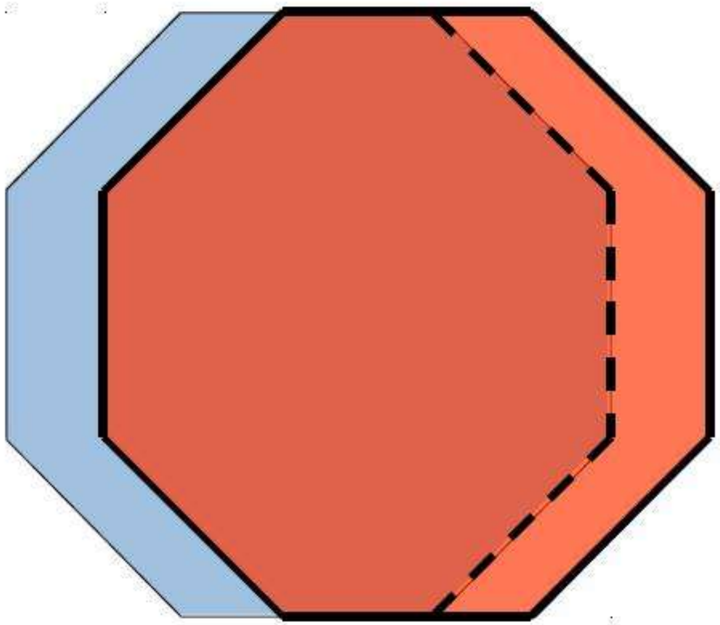
Для точки стены (k, t) отдельно подсчитаем шурупы

- в столбце над точкой
- в строке левее точки
- по диагонали влево-вверх от точки
- по диагонали вправо-вверх от точки



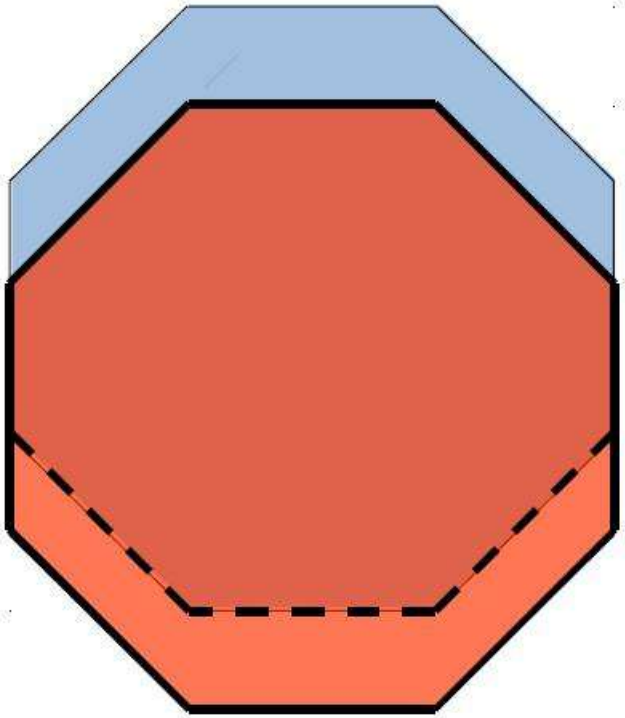
2. Для описания внутренней части картины находим 8 граничных отрезков: вертикальные, горизонтальные и диагональные.
3. Подставляем картину в каждую возможную точку стены (k, t). В B[1..N][1..M] храним количество закрытых шурупов.
   a. Подставляем левую верхнюю точку картины в точку (1,1). B[1][1] := количество закрытых шурупов.

б. Сдвигаем картину по одной клетке вправо вдоль верхней линии. Вычитаем из предыдущего результата открывающиеся шурупы по вертикали и диагоналям с левой стороны от картины.



Добавляем закрывающиеся шурупы по вертикали и диагоналям.

в. Сдвигаем картину по одной клетке вниз. Вычитаем из предыдущего результата открывающиеся по горизонтали и диагоналям с верхней стороны от картины.



Добавляем закрывающиеся шурупы по горизонтали и диагоналям.

Сложность *2-го способа* – O(*RC*+*NM*)