

# Задача 1

## Постановка

Кот задался целью поймать как можно больше мышей. Мыши обитают в  $n$  норках, которые соединены  $n - 1$  переходами. От одной норки до другой существует только один путь. Норка №1 является входом в мышиный лабиринт.

В результате перепланировки мышиного лабиринта было принято решение сделать переходы односторонними так, чтобы от входа в лабиринт (норка №1) можно было добраться до любой норки.

Когда кот залез в мышиный лабиринт, в  $i$ -ой норке находилось  $k_i$  мышей. Погоня началась. Каждая мышка, находящаяся в норке, из которой есть переход в другую норку, выбирает любой из таких переходов и переходит по нему. После этого кот аналогичным образом выбирает любой переход из той норки, в которой он сейчас находится, и переходит по нему. Так повторяется до тех пор, пока кот не окажется в норке, из которой никуда перейти нельзя. Кот поймает всех мышей, которые оказались в этой норке (и никуда не могут убежать).

Кот хочет поймать как можно больше мышей, а мыши хотят спасти как можно больше своих собратьев. В лабиринте работает мышиное радио, поэтому и кот, и мыши знают, сколько мышей находится в норках в любой момент времени. Мыши работают сообща. Если и кот, и команда мышей действуют оптимально, сколько мышей сможет поймать кот?

## Входные данные

В первой строке содержится целое число  $n$  – количество норок в мышином лабиринте. Во второй строке содержится  $n - 1$  целых чисел  $b_2, \dots, b_n$ , обозначающие существование перехода из норки  $b_i$  в  $i$ , при этом ( $1 \leq b_i < i$ )

В третьей строке находятся целых чисел  $k_1, \dots, k_n$ , означающие количество мышей в каждой норке в тот момент, когда кот решил залезть в мышиный лабиринт ( $0 \leq k_i \leq 10^9$ )

## Выходные данные

Необходимо вывести одно число – количество пойманных котом мышей при оптимальном поведении кота и мышиной команды.

## Пример 1

Входные данные	Выходные данные
3 1 1 2 0 0	1

## Пример 2

Входные данные	Выходные данные
2 1 3 0	3

## Задача 2

### Постановка

Курьеру необходимо перевезти ценный груз со склада до клиента с минимальными затратами.

Кратчайший путь от склада до дома клиента – прямое расстояние между координатами зданий. Однако не всегда такой путь возможен: где-то между складом и клиентом находится большой город, движение по которому затрудняется наличием пробок.

Перемещаться курьер может только по маршрутным точкам. Точка является маршрутной, если она принадлежит прямой, соединяющей склад и дом клиента, или находится на границе города (кольцевая автодорога).

Стоимость бензина для курьера составляет 1 у.е. за каждую единицу расстояния, пройденную по маршрутным точкам. Также курьер может ехать через город, однако тогда к стоимости бензина прибавится налог на передвижение, и тогда стоимость за каждую единицу расстояния составит 2 у.е.

Возможно перемещение курьера по границе города, за такое передвижение налог не взимается.

Итак, имеется карта местности от склада до дома клиента. Необходимо найти минимальные затраты на путешествие. Курьер выезжает со склада с координатами  $(x_S, y_S)$  и должен попасть домой к клиенту с координатами  $(x_E, y_E)$ . Очевидно, что координаты всегда различны – клиент не может жить на складе.

Город является выпуклым многоугольником, также известно, что никакие три точки, ограничивающие город, не лежат на одной прямой. Склад и дом клиента не находятся ни внутри города, ни на его границе. Точки, ограничивающие город, заданы в порядке против часовой стрелки.

### Входные данные

В первой строке даны 4 целых числа:  $x_S, y_S, x_E, y_E$ , являющиеся координатами склада и дома клиента.

Во второй строке дано целое число  $n$  – количество точек в многоугольнике, ограничивающем город.

В третьей строке даны  $n$  пар целых чисел – координаты  $(x, y)$  различных точек многоугольника, задающего город.

### Выходные данные

Минимальная возможная стоимость поездки курьера со склада до клиента.

### Пример 1

Входные данные	Выходные данные
0 3 10 3 3 1 0 9 0 5 3	10.000000

## Пример 2

Входные данные	Выходные данные
-90 50 90 50 3 0 0 1 100 -1 100	181.000000

## Задача 3

### Постановка

Робот Анатолий гуляет по пиксельному полю экрана. Его задача – добраться от левого верхнего края экрана до правого нижнего, учитывая, что по битым пикселям перемещаться нельзя. Шагать можно из текущего пикселя вниз, вверх, вправо и влево. Наискосок ходить нельзя. Известно, что левый верхний пиксель не является битым.

Экран представлен матрицей размера  $n * m$ .  $k$  клеток матрицы не являются битыми пикселями, и робот может по ним пройти. Анатолий может пожертвовать одной единицей своего быстродействия, чтобы временно починить все пиксели в одной строке или столбце. За каждый новый временно починенный столбец или строку придется платить заново.

Чтобы починить столбец или строку пикселей, роботу необходимо стоять на пикселе, который изначально не сломан. Как только Анатолий убирает временное восстановление пикселей со строки или столбца, все клетки этого набора становятся битыми.

Необходимо найти минимальное число единиц быстродействия робота Анатолия, которыми он должен пожертвовать для достижения своей цели.

### Входные данные

В первой строке находятся три целых числа  $n, m, k$ . Следующие  $k$  строк задают различные пары координат подсвеченных клеток:  $(r_i, c_i)$

### Выходные данные

Минимальное число единиц быстродействия, необходимых для достижения нижнего правого угла экрана. Если это невозможно, вывести -1.

### Пример 1

Входные данные	Выходные данные
7 9 7 1 1 7 8 3 2 5 3 2 8 4 8 3 6	2

## Пример 2

Входные данные	Выходные данные
5 10 5 1 1 3 2 5 8 3 6 1 8	3

## Пример 3

Входные данные	Выходные данные
9 10 3 1 1 6 6 8 4	-1