
Разбор задачи «Огород Марио»

В задаче требовалось найти клетку внутри прямоугольника, у которой минимальное из манхэттенских расстояний до данного набора клеток максимален. Напомним, что Манхэттенское расстояние между клетками с координатами (x_1, y_1) и (x_2, y_2) — это величина $|x_1 - x_2| + |y_1 - y_2|$.

Подзадача 1

В случае, когда изначально заражена только одна клетка, требуется найти внутри прямоугольника клетку с максимальным манхэттенским расстоянием до данной клетки. Можно показать, что такой клеткой всегда будет являться один из углов прямоугольника. Таким образом, для решения данной подзадачи достаточно было найти максимальное манхэттенское расстояние от данной клетки до одного из углов прямоугольника.

Подзадачи 2 и 3

В этих подзадачах достаточно было промоделировать процесс, описанный в условии задачи. Баллы, набираемые решением, зависят от эффективности реализации моделирования. Также для решения этих подзадач можно было перебрать все клетки внутри прямоугольника и для каждой из них найти минимальное манхэттенское расстояние до одной из данных клеток. Асимптотика времени работы такого решения составляет $O(nrc)$.

Подзадачи 4 и 6

Подзадачи, в которых ограничения на размер прямоугольника позволяют это сделать, можно применить поиск в ширину. Построим граф, вершинами которого будут клетки, а ребром будут соединены все пары клеток, соседних по стороне. В этом графе запустим алгоритм поиска в ширину одновременно из всех изначально зараженных клеток (для этого достаточно все эти клетки изначально поместить в очередь, используемую при обходе). Теперь для каждой клетки мы знаем расстояние до ближайшей из изначально зараженных. Чтобы получить ответ на задачу, выберем из всех этих расстояний максимальное. Асимптотика времени работы такого решения при эффективной реализации поиска в ширину составляет $O(rc + n)$.

Полное решение

Очевидно, что если в какой-то момент все клетки были заражены, то в каждый из последующих моментов они также все заражены. Значит, в задаче можно применить двоичный поиск по ответу. Таким образом, задача сводится к следующей: проверить, верно ли, что через t секунд после начала процесса все клетки были заражены.

Рассмотрим для каждой изначально зараженной клетки множество всех клеток внутри прямоугольника, манхэттенское расстояние до которых от этой клетки не превосходит t . Назовем такое множество *окрестностью* клетки. Легко заметить, что множество клеток, зараженных через t секунд после начала процесса — это объединение всех окрестностей изначально зараженных клеток. Заметим также, что каждая окрестность выглядит как квадрат, повернутый относительно сторон исходного прямоугольника на 45° , у которого расстояние от центра до углов составляет t .

Чтобы проверить, что объединение окрестностей полностью покрывает прямоугольник, применим преобразование ко всем клеткам: клетку с координатами (x, y) заменим на точку с координатами $(x + y, x - y)$. Так как сумма и разность чисел всегда имеют одну четность, у получившихся точек четность координат будет совпадать. Посмотрим, во что переходит окрестность клетки при таком преобразовании. Можно понять, что она переходит во все точки внутри некоторого квадрата, у которых одинаковая четность координат. Клетки исходного прямоугольника же переходят в точки с одинаковой четностью координат, расположенные внутри некоторого прямоугольника, повернутого относительно осей на 45° . Для простоты мы можем считать, что квадраты содержат в себе не только точки с одинаковой четностью координат, но и все остальные, ведь добавление этих точек никак не поспособствует покрытию необходимого нам прямоугольника, так как всего его точки имеют одинаковую четность координат.

Эта задача похожа на задачу объединения прямоугольников, которая решается с помощью метода сканирующей прямой. Применим схожий метод: запустим сканирующую прямую слева направо и будем следить за событиями, которые происходят. Событием является либо начало (левая сторона) какого-то из квадратов, либо его конец (правая сторона). Для каждой y -координаты будем хранить, сколькими квадратами покрыта сейчас точка сканирующей прямой с соответствующей координатой. Когда сканирующая прямая встречает начало квадрата, нужно для всех y -координат,

попадающих в этот квадрат, увеличить их счетчики на единицу, а когда прямая встречает конец квадрата — уменьшить. В некоторые моменты времени необходимо проверять, что все точки с текущей x -координатой, лежащие в прямоугольнике, покрыты хотя бы один раз. Можно доказать, что достаточно осуществлять такую проверку только в положениях сканирующей прямой, x -координата которых отличается от x -координаты какого-то из квадратов не более, чем на 1. Таким образом, есть $O(n)$ положений сканирующей прямой, в которых достаточно осуществить проверку. Каждая проверка заключается в следующем: надо проверить, что внутри какого-то отрезка y -координат все точки с фиксированной четностью покрыты хотя бы один раз. Границы этого отрезка определяются пересечением интересующего нас прямоугольника и сканирующей прямой и могут быть вычислены за константное время. Четность y -координат точек внутри отрезка, интересующих нас, определяются четностью текущей x -координаты: эти четности должны совпадать.

Таким образом, на сканирующей прямой мы хотим делать следующие операции:

- Увеличить все числа на отрезке на единицу.
- Уменьшить все числа на отрезке на единицу.
- Проверить, есть ли среди точек с фиксированной четностью внутри отрезка хотя бы одна, значение в которой нулевое.

Так как значения в точках никогда не могут стать отрицательными, для таких операций подойдет дерево отрезком на минимум с прибавлением на отрезке. Для того, чтобы правильно обрабатывать точки нужной четности, можно использовать два дерева отрезков: одно для четных y -координат, другое для нечетных. Тогда каждая операция сводится либо к прибавлению на отрезке, либо к взятию минимума на определенном отрезке в каждом из деревьев.

Для того, чтобы получить полное решение, необходимо также применить метод сжатия координат, так как исходные координаты могут быть слишком большими для того, чтобы сохранить их все в дереве отрезков. При сжатии координат нужно очень внимательно отнестись к тому, что нас иногда интересуют точки определенной четности. Это значит, что нас могут интересовать не только y -координаты всех вершин квадратов, но и другие y -координаты, расположенные между ними и имеющие нужную для нас четность. Самым простым способом решения этой проблемы является рассмотрение при сжатии координат не только всех y -координат квадратов, но и всех y -координат, отличающихся от них не более, чем на 2.

Двоичный поиск по ответу работает за время $O(\log C)$, где $C = \max(r, c)$. Проверка с помощью сканирующей прямой и дерева отрезков работает за время $O(n \log n)$. Таким образом, асимптотика времени работы всего решения составляет $O(n \log n \log C)$. Для получения полного балла по задаче, несмотря на небольшие ограничения, необходимо было довольно эффективно реализовать решение. Частичные баллы можно было набрать, не применяя сжатие координат, либо реализовав дерево отрезков «наивно».