Разбор задачи «Возрастающий массив»

Решим задачу жадным алгоритмом. Рассмотрим по-очереди элементы массива от 1-го до n-го. Каждый раз будем из двух вариантов выбирать тот, который во-первых, больше либо равен предыдущего выбранного значения, а во-вторых, если все равно оба варианта возможны, то меньший их двух (то есть, отрицательный). Если на каждом шаге получилось выбрать очередной элемент, то решение найдено. Иначе, решения не существует.

Разбор задачи «Перлы и конвертер»

Чтобы решить эту задачу заведем k очередей для каждого цвета жемчужин, а также будем поддерживать количество непустых из них. Тогда, если при добавлении очередной жемчужины в нужную очередь, все очереди становятся непустыми, соберем набор из жемчужин находящихся в началах очередей.

Важно поддерживать актуальные данные в очередях, поэтому при добавлении i-й жемчужины, важно не забыть вынуть $i - m - 1$-ю из ее очереди.

Обратим внимание, что каждую жемчужину мы ровно один раз добавим в очередь и ровно один раз вынем, а также проход по очередям будет совершен только в случае нахождения ответа. Получаем решение за $O(n + k)$.

Разбор задачи «Новый корабль»

Для того, чтобы решить первую подзадачу, переберем размер креста и его левый-верхний угол центрального квадрата креста и проверим, что каждая из клеток зафиксированного креста пригодна для строительства. Такое решение работает за $O(n^3 \cdot m^2)$и проходит первую подзадачу.

Улучшим данное решение, заметим что если мы зафиксировали левый-верхний угол центрального квадрата креста и можно построить крест размера k с ним, то с ним можно построить и крест любого меньшего размера, так как множество креста меньшего размера будет подмножеством клеток креста большего размера с ним. А значит будет работать такое решение переберем левый-верхний угол центрального квадрата креста и будем пытаться увеличить ответ, пока это возможно, проверяя, что каждая из клеток зафиксированного креста пригодна для строительства, так же, как в прошлой подзадаче. Такое решение работает за $O(n^2 \cdot m^2)$ и проходит первые 2 подзадачи.

Улучшим первое решение по-другому, заметим, что проверять то, что каждая из клеток зафиксированного креста пригодна для строительства, равносильна проверке, что число клеток пригодных для строительства в каждом из 5 квадратов креста равно квадрату размера креста. Это легко проверяется за $O(1)$ с помощью префиксных сумм. Такое решение работает за $O(n^2 \cdot m)$. И проходит первые 3 подзадачи.

Скомбинировав идеи предыдущих двух подзадач получаем решение, работающее за $O(n \cdot m)$, которое проходит все тесты.

Разбор задачи «Новый фонтан»

Построим граф, вершинами которого будут единичные квадратики, из которых состоит основание фонтана, а ребром соединены соседние по стороне квадратики. Заметим, что высота столба воды на каждом столбике определяется тем, насколько высокий есть «барьер», который не дает воде на этом столбике стекать за край поля. Более формально, суммарная высота столбика и столба воды на нем равна минимально возможному числу x, такому что существует путь от соответствующего единичного квадратика до какого-то единичного квадратика на границе, и высоты всех столбиков на этом пути не превышают x. Все решения жюри в какой-то форме используют это утверждение.

Подзадачи 1 и 2

В первых двух подзадачах на столбиках высоты 2 точно не могло быть воды сверху, а на столбиках высоты 1 сверху был столб воды высоты 1 в том и только в том случае, если нет пути от этого столбика до края по столбикам высоты 1.

Для решения первой подзадачи можно было любым алгоритмом обхода графа, например, обходом в глубину, для каждого столбика высоты 1 проверить, существует ли такой путь. Ответом является количество столбиков, для которых такого пути нет. Обход в глубину на таком графе работает за время, пропорциональное nm, всего будет не более nm запусков. Таким образов, асимптотика времени работы решения составляет $O(n^2m^2)$.

Для решения второй подзадачи достаточно было заметить, что достаточно лишь выделить в графе компоненты связности, состоящие из столбиков высоты 1. Это можно сделать всего одним запуском алгоритма обхода графа, а ответом будет являться суммарный размер всех компонент связности, в которых есть хотя бы один квадратик, расположенный на границе. Асимптотика времени работы этого решения составляет $O(nm)$.

Подзадача 3

Для нахождения высоты столба воды на каждом столбике воспользуемся двоичным поиском. Чтобы проверить, что суммарная высота столбика и столба воды на нем не превышает x, запустим поиск в глубину, который будет посещать только столбики, высота которых не превышает x. Если поиск в глубину находит путь до края, то суммарная высота столбика и столба воды на нем не превышает x, в противном случае она строго больше x. Асимптотика времени работы этого решения составляет $O(n^2m^2 \log H)$, где H — это максимальная высота столбика.

Подзадача 4

Для решения этой подзадачи достаточно было заметить, что суммарная высота столбика и столба воды на нем всегда равна высоте какого-то из столбиков. Таким образом, можно для каждой возможной высоты столбиков найти поиском в глубину компоненты связности столбиков, высота которых не превышает этой высоты, и прибавить к ответу суммарный размер компонент, не содержащих клетки границы. Асимптотика времени работы такого решения составляет $O(nmH)$.

Подзадача 5

Для решения задачи на полный балл воспользуемся алгоритмом Дейкстры. Заметим, что мы хотим для каждой клеточки найти путь от границы, максимальная высота столбика на котором минимальна. Алгоритм Дейкстры корректно работает в случае, когда весом пути яввляется не сумма весов ребер на нем, а максимальный из этих весов. Таким образом, достаточно лишь запустить алгоритм Дейкстры одновременно из всех клеточек границы, считая весом каждого ребра максимальную из высот двух его концов. Тогда после работы алгоритма найденное расстояние до каждого столбика будет равняться суммарной высоте этого столбика и столба воды на нем. При эффективной реализации алгоритма Дейкстры (с использованием кучи, set или TreeSet), асимптотика времени работы данного решения составляет $O(nm \log nm)$.