Индивидуальная олимпиада по информатике и программированию. Заключительный этап.

25 марта 2018 года

Задача А
Подземелье для принцесс

Постановка задачи

- Есть камеры вдоль коридора
- Вход находится между некоторыми двумя из них
- Боузер совершает вылазки по похищению b_j принцесс
- Нужно для каждой вылазки определить ближайшую ко входу камеру с вместительностью равной b_j или большей, если таковой нет

Решение

- Напишем ровно то, что просят в условии
- Найдем ближайшую камеру после входа с равной b_j вместительностью
- Найдем ближайшую камеру перед входом с равной b_j вместительностью
- Разберем случаи

Решение

- Если не нашли ни одну камеру с равной b_j вместительностью:
  - Ищем ближайшую с большей вместительностью после входа
  - Ищем ближайшую с большей вместительностью перед входом
  - Выбираем из них ближайшую или с меньшим номером

Комментарий

- Для удобства реализации вместительность уже занятых камер можно присваивать *0*, тогда их уже никогда не выберут

- Решение будет работать за $O(nm)$
- Частичные решения устроены аналогично, но чуть проще в реализации

Задача B
Ландшафтный
дизайн

Постановка задачи

- Есть последовательность чисел и число k
- За одно действие можно изменить элемент на 1
- За минимальное количество действий получить последовательность одного из двух видов:
  - $\{b, b + k, b, b + k, \dots\}$
  - $\{b, b - k, b, b - k, \dots\}$

Решение (31 или 84 балла)

- Будем решать для каждого случая отдельно
- Пусть конечная последовательность будет иметь вид $\{b, b + k, b, b + k, ...\}$
- Переберем значение b, $|b| \le 3000$
- За время $O(n)$ или $O(a)$ посчитаем число действий
- Асимптотика – $O(na)$ или $O(a^2)$

Полное решение (100 баллов)

- Прибавим ко всем элементам с нечетными номерами число k: $c = \{a_1 + k, a_2, a_3 + k, \ldots\}$
- Сделать из *a* последовательность нужного вида – то же самое, что сделать из *c* последовательность из равных чисел
- Нужно решить задачу при $k = 0$

Полное решение (100 баллов)

- Можно искать оптимальное значение *b* троичным поиском
- Посмотрим, как при изменении значения изменяется число действий
- Можно доказать, что оптимальное значение – медиана последовательности *c*
- При зафиксированном *b* суммарное число операций вычисляется, как сумма модулей попарных разностей
- Асимптотика – $O(n \log n)$

Задача C
Гонка

Постановка задачи

- Есть n персонажей, у каждого есть популярность b_i
- Есть m заездов, известны их результаты
- За i-е место в заезде дается a_i баллов
- Общий балл – сумма популярности и очков за заезды, кроме s минимальных
- Найти минимальное количество дополнительных заездов, чтобы персонаж v обогнал персонажа u по общему баллу

Общие замечания

- Очевидно, что в каждом дополнительном заезде v должен будет занимать первое место, а u – последнее
- Если $b_u = b_v = 0$, то понадобится не более $m + 1$ дополнительных заездов

Решение (от 12 до 58 баллов)

- По очереди будем добавлять дополнительные заезды
- Каждый раз будем пересчитывать общие баллы
- Чтобы заново не искать s минимальных баллов за заезды, будем запоминать их
- Множество минимальных баллов при добавлении дополнительного заезда меняется не более, чем на 1 элемент
- Асимптотика – $O(qm^2s)$, $O(qm^2 \log m)$ или $O(qms)$

Полное решение (100 баллов)

- Двоичный поиск по ответу
- Пусть было x дополнительных заездов
- Сумма всех баллов – исходный общий балл, увеличенный на xy, где y – количество очков за каждый дополнительный заезд
- Нужно выбирать s минимальных элементов среди имеющихся m результатов заездов и числа y, взятого x раз

Полное решение (100 баллов)

- Кандидатами на эти s минимумов являются s минимальных баллов за первые m заездов и s копий числа y
- Воспользуемся сортировкой или методом двух указателей: каждый раз брать минимального из очередных кандидатов
- Асимптотика – $O(qs \log C)$

Задача D
Подарок для Луиджи

Постановка задачи

- Есть четыре палочки
- Можно разломать каждую из них на несколько новых палочек
- Нужно разломать палочки так, чтобы из них можно было выбрать четыре, из которых можно составить прямоугольник с наибольшей возможной площадью

Решение (24 балла)

- Перебираем длины сторон
- Проверяем, что можно получить четыре палочки нужных длин
- Сложность $O(L^2)$

Решение (52 балла)

- Перебираем длину одной стороны
- Наибольшую возможную длину второй стороны ищем двоичным поиском
- Проверяем, что можно получить четыре палочки нужных длин
- Сложность $O(L \log L)$

Решение (74 балла)

- Проанализируем ответ
- Есть пять случаев из каких исходных палочек получатся его стороны:
    - Все четыре стороны получены разломом одной палочки
    - Каждая сторона получена из отдельно взятой палочки
    - Две стороны получены из одной палочки, две из второй
    - Три стороны получены из одной палочки и одна из второй
    - Две стороны получены из одной палочки, остальные две из еще двух

Решение (74 балла)

- Рассмотрим все случаи
- Для каждого случая рассмотрим случаи взаимного расположения сторон (смежные или противоположные)
- Сложность *O(1)*

Полное решение (100 баллов)

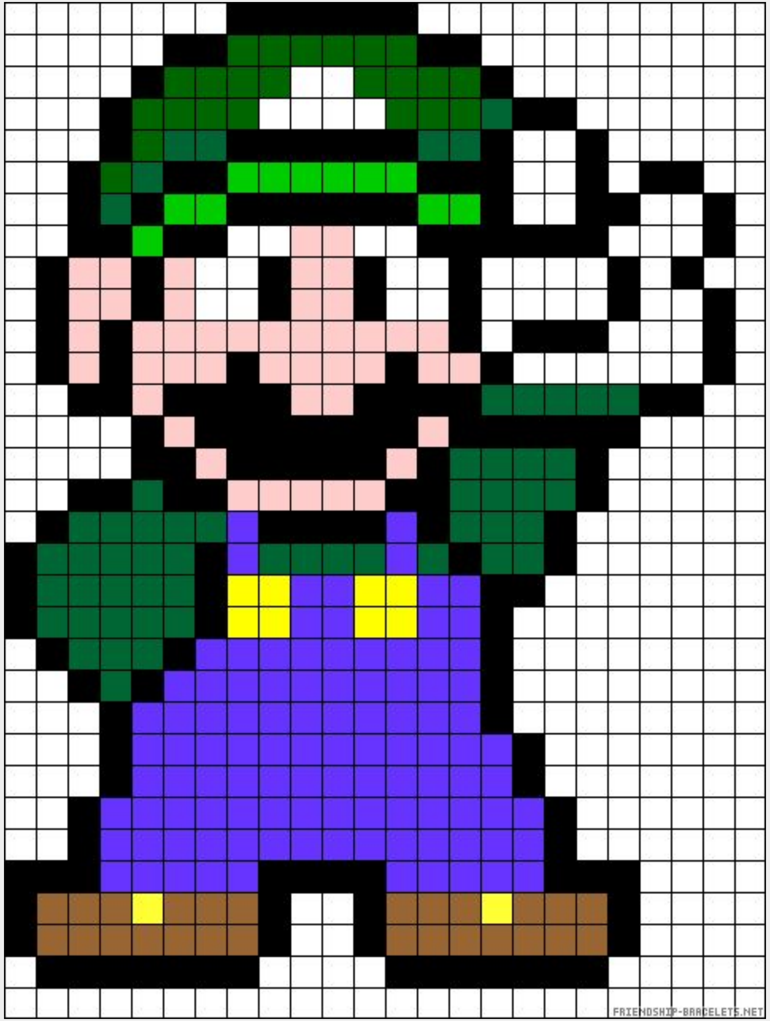
- Добавим к предыдущей идее корректную обработку больших длин
  - Длинная арифметика

Частичные решения (от 0 до100 баллов)

- Так как все тесты оценивались раздельно, можно было рассмотреть не все случаи и получить некоторое число баллов

Задача E

Марио и параллельный мир



Постановка задачи

- Есть прямоугольное клетчатое поле
- В каждой клетке написаны числа (очки)
- Марио хочет дойти из левой нижней клетки в правую верхнюю так, чтобы набрать в сумме наибольшее возможное число очков.
- Нужно занять какую-то клетку так, чтобы Марио, действуя оптимально, получал бы наименьшее возможное количество очков

Решение (29 баллов)

- Переберем клетку, которую будем занимать
- Посчитаем максимальное число очков, которые можно теперь набрать, с помощью динамического программирования
- Выберем минимальное
- Данное решение работает за $O(n^2m^2)$

Решение (61 балл)

- Найдем в изначальном поле какой-нибудь оптимальный путь
- Нет смысла занимать клетку, которая в него не входит
- В предыдущем решении будем пытаться занять клетки только из найденного оптимального пути
- Данное решение работает за $O((n + m)nm)$

Решение (100 баллов)

- Посмотрим на оптимальный путь
- Он проходит ровно один раз через каждую побочную диагональ (множество клеток с одинаковой суммой координат)
- Если займем клетку на оптимальном пути, то новый путь Марио пройдет через некоторую другую клетку на этой побочной диагонали

Решение (100 баллов)

- Через какую?
- Через ту, сумма у которой сумма максимального количества очков, которые можно получить на пути из начальной клетки в нее и из нее в конечную клетку, наибольшая среди всех таких сумм для клеток на данной диагонали (кроме занятой)
- Если предподсчитать слагаемые, можно находить эту сумму за *O(1)*

Решение (100 баллов)

- Предподсчитаем при помощи этих сумм для каждой побочной диагонали по две клетки с наибольшими суммами очков, которые можно получить пройдя через них
- Минимальный среди вторых максимумов будет ответом на задачу
- Итоговая сложность $O(nm)$

Задача F
Огород Марио

Постановка задачи

- Есть прямоугольное поле
- Некоторые клетки заражены
- За секунду все клетки, соседние с зараженными, заражаются
- Через сколько секунд заразится весь прямоугольник?

Решение (10 и 15 баллов)

- *$n = 1$*: ответ – максимальное манхэттенское расстояние от клетки до одного из углов
- Промоделируем процесс «наивно»
- Переберем клетку, для каждой найдем минимальное из манхэттенских расстояний до изначально зараженных. Выберем из этих величин максимум
- Асимптотика – $O(nrc)$

Решение (35 баллов)

Для каждой клетки найдем минимальное из манхэттенских расстояний до зараженных

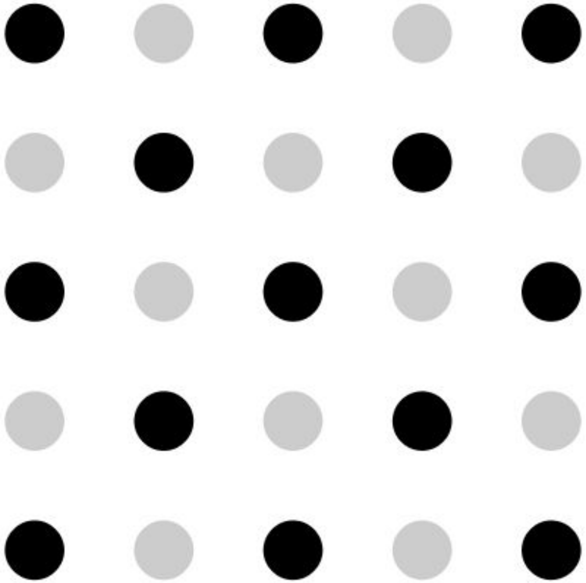
- Запустим поиск в ширину из всех зараженных клеток одновременно
- Для этого изначально все эти клетки положим в очередь
- Ответ – максимальное из расстояний до всех клеток
- Асимптотика – $O(rc + n)$

Полное решение (100 баллов)

- Двоичный поиск по ответу
- Пусть прошло *t* секунд с начала процесса
- Область на которую распространилось заражение из одной клетки – окрестность
- Окрестность представляет из себя квадрат, повернутый на *45°*, расстояние от центра до угла *t*
- Зараженная территория – объединение всех окрестностей

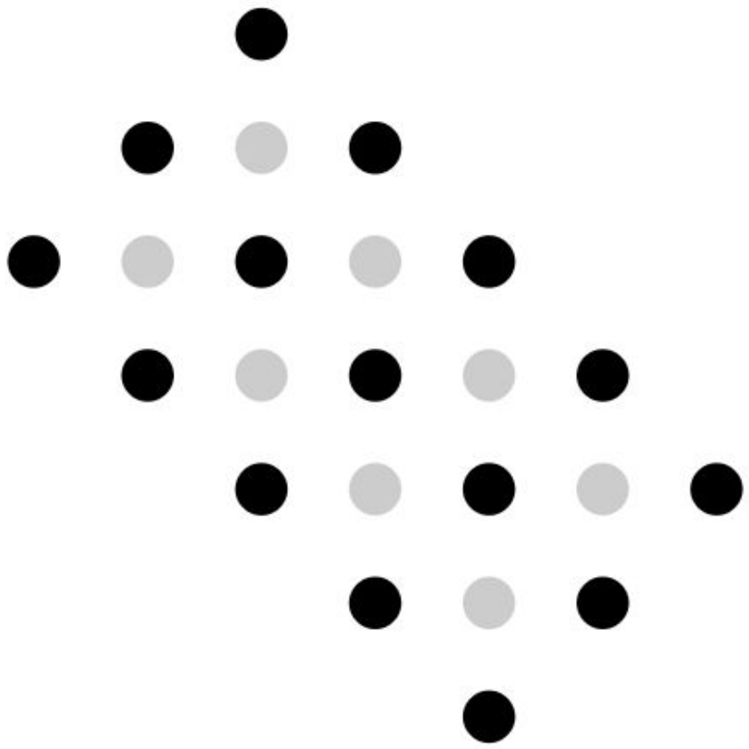
Полное решение (100 баллов)

- Повернем плоскость на *45°*: заменим клетку *(x, y)* на точку *(x + y, x - y)*
- Окрестности стали квадратами с пропусками:



Полное решение (100 баллов)

- Исходное поле стало прямоугольником, повернутым на *45°*, с пропусками:



Полное решение (100 баллов)

- Хотим проверить, что квадраты с пропусками покрывают прямоугольник с пропусками
- Пропуски в квадратах можно убрать, они ни на что не влияют
- Решим задачу с помощью сканирующей прямой
- Для каждой точки прямой храним, сколько раз она покрыта квадратами

Полное решение (100 баллов)

- События: начала и концы квадратов
- Когда встречаем событие, прибавляем или вычитаем единицу на отрезке
- Чтобы проверить, что все точки прямоугольника с текущей x-координатой покрыты, надо проверить, что минимум на отрезке среди значений с индексами фиксированной четности строго положителен

Полное решение (100 баллов)

- Проверку имеет смысл делать только в *x*-координатах, которые отличаются от *x*-координаты какого-то квадрата не более, чем на *1*
- Храним все счетчики в дереве отрезков с прибавлением на отрезке и минимумом на отрезке
- Для удобства разделения четностей можно использовать два дерева отрезков

Полное решение (100 баллов)

- Координаты большие – необходимо сжатие
- Сжимать надо аккуратно: бывают точки, которые не являются y-координатами никакого из квадратов, но необходимы нам, так как имеют нужную четность
- Простой выход – для каждой координаты квадрата y следить за y, $y + 1$, $y + 2$

Полное решение (100 баллов)

- Всего $O(n)$ событий и проверок, каждое событие и проверка за время $O(\log n)$
- Асимптотика всего решения – $O(n \log n \log C)$
- Требовалась эффективная реализация
- Можно было не сжимать координаты или вместо дерева отрезков использовать массив

print(’Спасибо за внимание’)