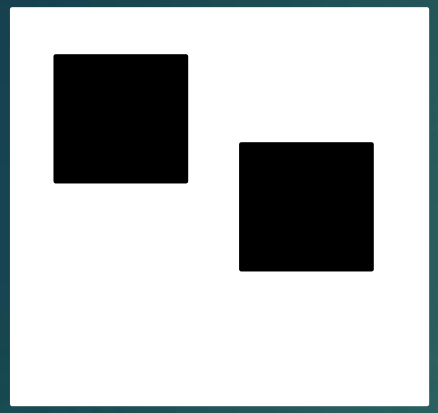
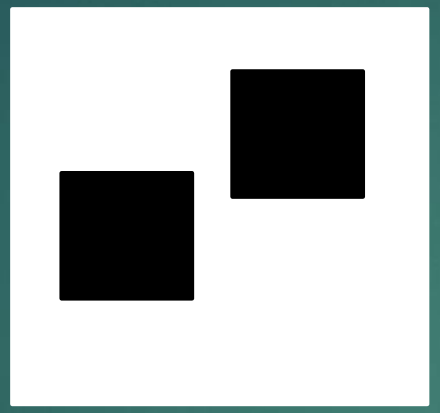
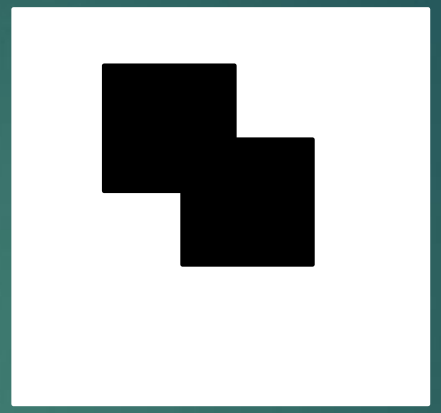
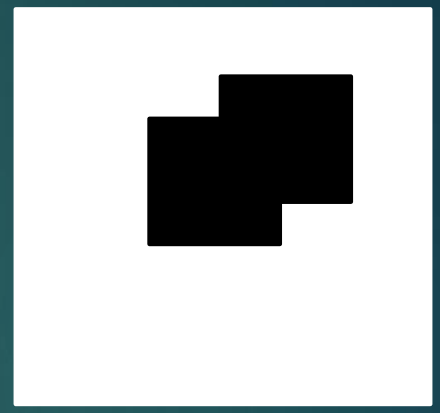
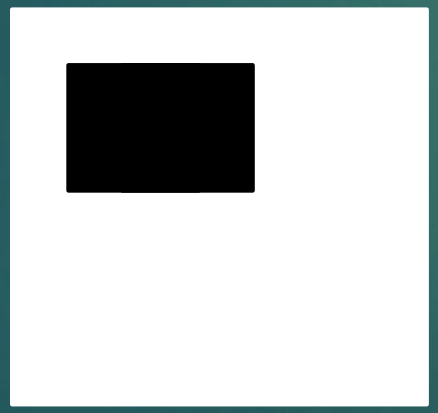
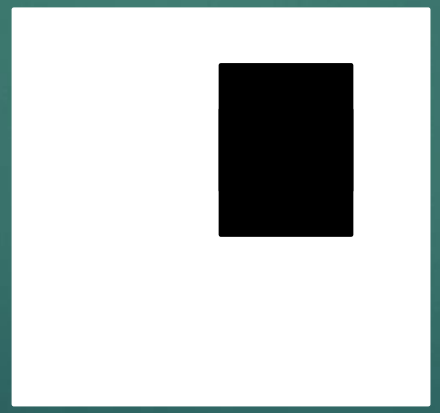
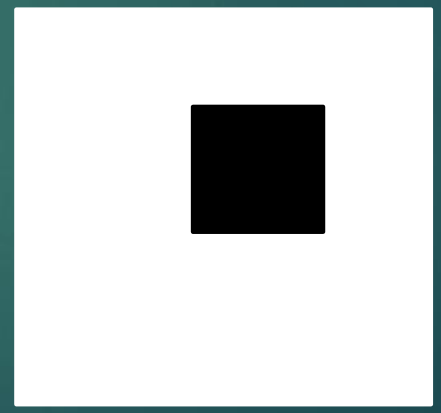
Разбор задач 1 тура 3-го этапа Республиканской олимпиады по информатике 2018 года

Тур 1 Задача 1
Два квадрата

Условие

- Дано поле, изначальное состоящее только из белых клеток, на котором нарисовали два квадрата черного цвета с одинаковой длиной стороны K.
- Требуется восстановить K и верхние левые углы нарисованных квадратов.

Возможные расположения квадратов

Частичные решения

- 1 подзадача. Проверим все возможные варианты расположения квадратов на поле (их всего 12: один для n = 1 и одиннадцать для n = 2).
- Подзадачи 2, 3, 4. Предполагаются решения, перебирающие длину стороны K и/или координаты клеток квадратов.
- В зависимости от оптимальности перебора такие решения набирают до 80 баллов.

Полное решение

- Найдем самую левую верхнюю клетку (x_1, y_1), покрашенную в черный цвет. Очевидно, что она будет являться левой верхней клеткой одного из квадратов.
- Аналогично найдем правую нижнюю клетку (x_2, y_2), покрашенную в черный цвет. Несложно заметить, что она обязательно будет являться правой нижней клеткой ***другого*** квадрата.
- Имея левую верхнюю клетку, можно вычислить размер нарисованных квадратов. Будем одновременно двигаться вправо и вниз из этой клетки, пока не наткнемся на белую клетку или на границу поля. Количество сделанных шагов и есть размер квадратов K.
- Сложность решения: $O(n^2)$

Тур 1 Задача 2
Творческие выходные

Условие

- N досок забора окрашиваются в K шагов с использованием K банок краски разных цветов
- На очередном шаге выбирается некоторый ранее не выбранный цвет и некоторые доски забора, каждая из которых перекрашивается в данный цвет
- На покраску одной доски уходит 1 минута
- Известны конечные цвета досок
- Определить максимальное количество времени, которое могло уйти на такую покраску

Частичные решения

- 1 подзадача: $n \leq 10, k \leq 2$. Так как цветов всего два, то есть только два порядка выбора цветов красок. Ответ равен $2 \cdot \max(a, b) + \min(a, b)$, где a и b – количества досок соответствующих цветов.
- 2 подзадача: $n \leq 100, k \leq 10$. Переберем все $k! = 1 \cdot 2 \cdot \ldots \cdot k$ вариантов порядка покраски и выберем лучший.
- 3 подзадача: $n \leq 1000, k \leq 15$. Ускорим решение подзадачи 2 с помощью динамического программирования по маскам.

Идея полного решения

- Заметим, что неважен порядок досок в заборе, важны лишь количества досок каждого цвета.
- Посчитаем количество досок каждого цвета в массив c.
- Пусть есть два цвета x и y и при этом $c_x < c_y$. Тогда всегда оптимально сначала красить доски в цвет x, а затем в y, так как это дает возможность покрасить на $c_y - c_x$ досок больше, чем наоборот
- Таким образом, оптимально выбирать цвета красок в порядке возрастания количеств досок такого цвета в конечной раскраске

Решение подзадач 4 и 5

- Отсортируем массив c по возрастанию
- Как посчитать ответ, зная порядок покраски?
- Пусть m – количество еще не обработанных досок. Изначально $m = n$
- Идем по массиву c. На каждом шаге увеличиваем ответ на m, а m уменьшаем на c_i.
- В зависимости от скорости сортировки решение проходит 4/5 подзадачу. Для получения 100 баллов на C++ можно использовать std::sort, на Pascal - реализовать какой-нибудь алгоритм эффективной сортировки, например, quicksort.
- Для последней подзадачи нужно не забыть 64-битные типы данных (long long или __int64 в C++, int64 в Free Pascal), так как ответ может не вместиться в обычный 32-битные int / longint
- Сложность решения: $O(K \log K + N)$

Тур 1 Задача 3
Непростая сумма

-

Условие

- Дан массив a из n чисел
- Необходимо вычислить сумму $a_i \bmod a_j$ по всем $1 \leq i, j \leq n$
- x mod y – остаток от деления x на y

Решение подзадачи 2

- Все числа различны, тогда массив a – перестановка чисел от 1 до n, то есть каждое число от 1 до n встречается ровно один раз.
- Переберем все y от 1 до n. Тогда необходимо узнать сумму x mod y по всем x от 1 до n.
- Заметим, что x mod y = x - $\left\lfloor \frac{x}{y} \right\rfloor \cdot y$, где $\lfloor a \rfloor$ – целая часть числа a.
  Например, 7 mod 3 = 7 - $\left\lfloor \frac{7}{3} \right\rfloor \cdot 3 = 7 - 6 = 1$.
- Легко видеть, что для фиксированного y количество различных значений $\left\lfloor \frac{x}{y} \right\rfloor$ не превосходит $\left\lfloor \frac{n}{y} \right\rfloor$ и, более того, если
- $\left\lfloor \frac{x}{y} \right\rfloor = 0$, то $0 \le x < y$

  $\left\lfloor \frac{x}{y} \right\rfloor = 1$, то $y \le x < 2y$

  $\left\lfloor \frac{x}{y} \right\rfloor = 2$, то $2y \le x < 3y$

  …

  $\left\lfloor \frac{x}{y} \right\rfloor = \mathrm{k}$, то $\mathrm{ky} \le x < (k+1)y$

Решение подзадачи 2

- Из наблюдений выше, все x, дающие одно и то же значение при делении на y, можно «объединить» в одну группу.
- Для заданного y переберем все возможные неполные частные k (от 0 до $\left\lfloor \frac{n}{y} \right\rfloor$)
- Тогда к ответу нужно прибавить все сумму x – k * y по всем x от k·y до min(n, (k+1)·y – 1), что равно сумме иксов в этом промежутке, минус их количество, умноженное на k · y
- Для упрощения вычислений можно заметить, что сумма по всем k значений сумм иксов, принадлежащих соответствующему промежутку, равна сумме чисел в массиве, которая для подзадачи 2 равна $\frac{n(n+1)}{2}$

Сложность решения

- Оценим сложность такого решения
- Для каждого y количество совершаемых операций равно O(n / y).
- Тогда все решение будет иметь сложность $O(\frac{n}{1} + \frac{n}{2} + \frac{n}{3} + \ldots + \frac{n}{n})$, что равно $O(n \log n)$ с очень маленькой константой.
- Гармонический ряд: $\sum_{i=1}^{n} \frac{1}{i} \approx \ln n$.

Полное решение

- Для полного решения необходимо посчитать массив c_i вхождений каждого числа
- Тогда имея некоторый отрезок иксов $[k \cdot y, \min(n, (k+1) \cdot y - 1)]$ нужно найти количество чисел в массиве из этого отрезка
- Чтобы сделать это за O(1), посчитаем массив префиксных сумм над массивом c_i.
- Все одинаковые числа в качестве y обработаем как одно, умножив ответ для него на c_y. Если делать для одинаковых чисел каждый раз заново, асимптотика может стать квадратичной.

Тур 1 Задача 4
Роборалли

Условие

- Дан граф, из каждой вершины которого исходит ровно одно ориентированное ребро
- На каждом ребре записана буква
- Игрок стартует из всех вершин
- Из каждой вершины игрок делает M шагов, при этом выписывая буквы на ребрах и тем самым формируя строку длины M
- Найти K-ую в лексикографическом порядке среди всех N полученных строк

Решение

- Пусть дано множество строк. Тогда присвоим им номера так, чтобы одинаковые строки получили одинаковые номера, а строка a, лексикографически меньшая строки b, получила номер меньший, чем номер строки b
- Такое разбиение назовем разбиением строк на классы эквивалентности
- Тогда K-ая лексикографически строка будет иметь K-ый по возрастанию номер в разбиении

Решение

- Используем метод двоичных подъемов, чтобы посчитать num[v][i] – номер класса эквивалентности, который получит строка, сформированная из вершины v через 2^i шагов.
- Для этого потребуется массив up[v][i] – вершина, в которую мы попадем из v через 2^i шагов
- Будем считать эти значения в порядке увеличения i (от 0 до log M)

Решение

- Пусть i = 0. Тогда легко видеть, что up[v][0] – это вершина, в которую ведет ребро из v.
- Если это ребро цвета B, то num[v][0] = 1, если G – 2, если R – 3
- Пусть i > 0. Путь из любой вершины v длины 2^i состоит из двух путей: пути из v длины 2^{i-1} и пути из up[v][i – 1] той же длины.
- Тогда легко видеть, что up[v][i] = up[*up[v][i-1]*][i-1]
- Чтобы пересчитать классы эквивалентности, отсортируем пары (num[v][i – 1], num[up[v][i-1]][i – 1]) лексикографически с помощью сортировки подсчетом
- Жадно разобьем все вершины на новые классы эквивалентности, используя отсортированный массив пар
- Можно заметить аналогию с построением суффиксного массива

Решение

- Имея посчитанные значения num[v][i] легко определить, какая строка будет K-ой лексикографически из строк длины M
- Разобьем M на сумму степеней двоек и точно таким же образом пересчитаем классы эквивалентности, используя массив num
- Тогда вершина, имеющая K-ый класс эквивалентности будет иметь K-ую лексикографически строку
- Сделаем M шагов из нее и выведем ответ
- Сложность решения: $O(N \log N)$.
- Можно было где-то неаккуратно написать и получить $O(N \log^2 N)$ - 80-100 баллов