Теоретическая часть

Задача 1.

Ответ на задачу: $\sum_{i=1}^{K} a[i]-(k-1)$. Первый фильтр даёт $a[1]$ розеток, а каждый из остальных на одну меньше, чем в нем имеется.

Задача 2.

Дополним заданную расстановку ладей дополнительными ладьями с координатами $(0, 0)$ и $(w+1, h+1)$, где w и h – размеры доски. Это позволит искать требуемый прямоугольник только в промежутках между ладьями.

Отсортируем ладей по возрастанию координаты x и рассмотрим разности $c_i=x_i-x_{i-1}-1$ – расстояние между соседними ладьями по горизонтали. Затем отсортируем ладей по возрастанию координаты y и найдём разности $r_i=y_i-y_{i-1}-1$ – расстояния между соседними ладьями по вертикали.

Пусть искомый прямоугольник имеет размеры $a \times b$ клеток, тогда $a \le max\ c_i$, $r \le max\ r_i$, но в королевстве есть прямоугольник $max\ c_i \times max\ r_i$, таким образом, он и является ответом.

В зависимости от используемой сортировки решения получали от 15 до 30 баллов. Переборные решения получали не более 10 баллов.

Задача 3.

Если у числа первое число ноль или последняя цифра четная, то разбить строку не получится. Найдем в строке все пары последовательных чисел, таких, что число с меньшим индексом нечетное, а число с большим индексом не равно нулю. Это можно сделать за один проход. Таким образом, мы находим все места, где можно провести границу между числами в разбиении. Ответ – количество таких позиций + 1.

В зависимости от количества действий решения получали от 15 до 30 баллов.

Задача 4.

Задача решается методом динамического программирования. $dp[i][j]$ – количество билетов, состоящих из i цифр и суммой j. База: $dp[0][0] = 1$. Переход в динамике: dp[i][j]= $\sum_{k=-9}^{9} dp[i-1][j-k],\ i = 1, \dots, n, j = 0..9n$. Данная формула получается следующим образом: получить сумму в -значном билете можно либо прибавив к числу j цифру k, либо отняв её. Теперь подсчитаем ответ, воспользовавшись подсчитанными в динамике значениями. Чтобы получить счастливый билет, нужно, чтобы сумма первых N цифр совпадала с суммой вторых N цифр. Различных сумм – $18n + 1$ (от $-9n$ до $9n$). Сумму в первой половине можно получить $dp[n][i]$ способами, во второй половине – также $dp[n][i]$ способами. Отсюда ответ на задачу - $\sum_{i=-9*n}^{9*n} (dp[n][i])^2$.

Решения, использовавшие перебор, получали не более 10 баллов.

Задача 5.

Заметим, что если p/q превращается в $p/(p+q)$ для левого сына и превращается в $(p+q)/q$ для правого сына, в левом сыне $p < q$ и в правом сыне $p > q$, таким образом, можно восстановить номер дроби с конца. Начинаем восстанавливать ответ с самого наименьшего бита числа. Если $p > q$, то мы переходим наверх, ставим данный бит номера равным единице и заменяем p на $p - q$. Если $p < q$, то переходим наверх, ставя данный бит номера равным нулю и заменяем q на $q - p$. Переводя получившееся двоичное число в десятичную систему счисления, получаем искомый номер дроби.

Переборные решения получали не более 10 баллов.

Практическая часть

Задача 1.

Из n шагов каждой ногой спортсмен делает k маленьких шагов и $(n-k)/2$ больших шагов, где k принимает все нечетные значения до $\left\lfloor \frac{n}{3} \right\rfloor$ для нечетного n и все четные значения до $\left\lfloor \frac{n}{3} \right\rfloor$ для четного n. Итого каждая нога совершает

$(k\ +\ (n-k)/2)\ =\ (n+k)/2$ шагов. Для заданного k существуют $C_{k+(n-k)/2}^{k}$ способов выбрать малые шаги для левой ноги, аналогично для правой ноги (C_n^k – число сочетаний из n по k). Таким образом, распределить малые шаги можно $\left(C_{k+(n-k)/2}^{k}\right)^2$ способами. Итоговое число способов: сумма $C_{k+(n-k)/2}^{k}$ по всем $k = 1, \dots, \left\lfloor\frac{n}{3}\right\rfloor$, где k – нечетное, в случае нечетного n, либо сумма $C_{k+(n-k)/2}^{k}$ по всем $k = 0, \dots, \left\lfloor\frac{n}{3}\right\rfloor$, где k – четное, в случае четного n.

Задача 2.

Построим граф, вершинами которого будут являться исследовательские станции, ребрами – телепорты. В заданном графе требуется найти каркас минимального веса. В силу ограничения на число вершин для полного решения задачи требуется применить алгоритм Краскала со структурой данных «система непересекающихся множеств». В графе очень много ребер, но большинство из них не нужно, достаточно взять несколько наименьших. Рассмотрим вершину u, тогда кратчайшее ребро, инцидентное ей, будет v, ближайшая по одной из координат. Для каждой вершины кратчайшее ребро будем искать следующим образом:

1. Рассмотрим каждую координату в отдельности;
2. Отсортируем точки по одной из координат;
3. Для каждой точки составим ребро из неё и её правого соседа (если такой есть). Таким образом, в графе будет не более $3N$ ребер;
4. Запускаем алгоритм Краскала над отобранными в граф ребрами.

Утверждение: таким образом будет получен ответ на поставленную задачу. В самом деле, каждую вершину графа нужно соединить с другой, очевидно, что выгодно соединять с ближайшей.

Задача 3.

Одним из решений задачи является поиск в ширину, где вершинами будут тройки (x, y, z), где x и y – координаты машины, z – количество бензина, оставшееся в баке. В данном графе следует установить, можно ли добраться из вершины $(1, 1)$ в вершину (N, M), передвигаясь по незапрещенным клеткам поля. Переходы из вершины совершаются следующим образом: в доступные точки переходим, теряя 1 литр бензина, в точки с заправками добираемся и доливаем 5 литров либо пока бак не наполнится. Тогда если достижима хотя бы одна из вершин (N, M, z), где $z = 0, \ldots, K$, то существует путь из вершины $(1, 1)$ в вершину (N, M). Из всех таких путей выберем кратчайший.