

# Разбор задач

## 1 Классическая лига

### 1.1 Задача F. Паша читает по-древнерусски

*Задача предполагает умение участника использовать массивы и строки. Самая простая задача классической лиги. Для решения задачи надо создать массив строк, записать перевод буквы в соответствующую ячейку этого массива и вывести ответ, как требовалось в условии.*

В случае использования языков программирования C++ или Python задача решается без проблем с использованием встроенных средств. Другой способ решения - считать строку с пробелами, а затем выделить индекс и знак тире, используя цикл while. Благодаря короткой длине строк, их можно было спокойно считывать в языке программирования Pascal.

### 1.2 Задача G. Главное – тактика

*Задача предполагает умение участника использовать метод динамического программирования или надеяться на удачу.*

К сожалению, все успешные решения этой задачи, которые были сданы во время тура, использовали маленькие ограничения в условии и базировались на различных жадных идеях.

Предполагаемое решение - трёхмерное динамическое программирование, где  $dp_{i,j,k}$  означает, возможно ли набрать среди первых  $i$  раундов команду номер один веса  $j$  и команду номер два веса  $k$ . Заметим, что состояния, где  $j > a + 10$  бесполезны, поэтому мы можем ограничить  $j$  и  $k$  сверху весом соответствующей головы Горыныча +10. Зная вес первых двух групп можно однозначно определить вес третьей и сравнить с необходимым.

### 1.3 Задача H. Подарок богатырей

*Задача предполагает умение участника использовать массивы и знание модульной арифметики или умения доказывать свойства деления чисел (или хотя бы верить в них).*

Жюри предполагало, что задача окажется одной из самых простых в классической лиге, но в итоге задачу во время тура никто не решил.

Для начала можно заметить, что ответ - это сумма чисел на каком-то подотрезке массива  $+3$  + остаток от произведения чисел на нем при делении на большое число. Участникам требовалось заметить, что брать весь массив почти никогда не оптимально, при достаточно большом размере массива. А значит, нужно проверить все подотрезки массива.

Для этого удобнее всего перебирать левую границу подотрезка, а, зафиксировав её, увеличивать правую границу по одному элементу, начиная от левой, пересчитывая сумму и произведение чисел. Произведение чисел получается очень большим, поэтому даже при использовании длинной арифметики это решение не укладывается в ограничение по времени. Для решения задачи надо заметить, что, если мы будем на каждом шаге заменять само произведение, на остаток от деления произведения на наше *простое русское число*, то результат не изменится. Если вы хотите в этом убедиться, то попробуйте явно на бумаге расписать и сравнить остаток произведения двух чисел и произведение двух остатков.

## 1.4 Задача I. Рапторы – Пашина крепость

*Задача предполагает знание участником уравнения окружности.*

Для упрощения задачи смести все координаты так, чтобы центр окружности находился в центре координат. Как проверить, что точка  $(x, y)$  лежит внутри окружности?

Сравним  $x^2 + y^2$  с радиусом. Если это число меньше либо равно радиуса в квадрате, то точка находится внутри круга, граница которого задается данной в условии окружностью. Оказывается, что для того, чтобы выпуклый многоугольник лежал внутри круга, достаточно, чтобы все его вершины лежали внутри круга.

Частой ошибкой при решении данной задачи было извлечение корня. Подобные решения могли не работать из-за проблем с точностью, поэтому надёжнее всего было сразу решать задачу в целых числах.

## 2 Золотая лига

### 2.1 Задача J. Командное взаимодействие

*Задача предполагает знание участником алгоритма поиска в глубину и опыта работы со строками.*

В данной задаче требовалось найти центроид дерева. Алгоритм для его поиска можно посмотреть, например [здесь](#) (по ссылке описана центроидная декомпозиция, но частью её построения является нахождение центроида).

Другой проблемой при решении задачи был формат входных данных. Прозе всего задача решается использованием языка Python или [stringstream-ов](#) в языке C++. Или можно считать строку с пробелом, используя, например, `getline`, а затем вручную разбить на слова по пробелу и перевести их из строк в числа.