

Задача А. Путешествие

Возможны три случая:

1. камень находится между Ханты-Мансийском и посёлком (в этом случае $B - A = Y - X$, расстояние до Ханты-Мансийска равно $A - X$);
2. камень находится между посёлком и рекой (в этом случае $B - A = X + Y$, расстояние до Ханты-Мансийска равно $A + X$);
3. камень находится между рекой и Сургутом (в этом случае $B - A = X - Y$, расстояние до Ханты-Мансийска равно $A + X$).

Задача В. Часы

Часовая стрелка движется на 30 градусов каждый час плюс каждые 12 минут на 6 градусов. Минутная сдвигается на 6 градусов каждую минуту. Находим абсолютное значение разности между этими двумя углами. Если получается значение более 180, берем другой, меньший угол.

Задача С. Голубятни

Для каждой птицы переберём голубятню, к которой она может долететь. Чтобы не было проблем с вычислением расстояния, будем сравнивать квадрат расстояния. Если птица имеет координаты x_a, y_a , а голубятня имеет координаты x_b, y_b , то квадрат расстояния считается по формуле $(x_a - x_b)^2 + (y_a - y_b)^2$.

Задача D. Разделение подарков

Мы хотим минимизировать $|\sum_{i=1}^K i - \sum_{i=K+1}^N i|$. Каждая из сумм является арифметическая прогрессией. Тогда после преобразования получим $|K \cdot (K + 1)/2 - (N + K + 1) \cdot (N - K)/2|$. Так как мы стремимся, чтобы это значение было как можно меньше, то можно решить уравнение $(K \cdot (K + 1) - (N + K + 1) \cdot (N - K))/2 = 0$ и сравнить значения выражения для ближайших к корню уравнения целых чисел. После раскрытия скобок получаем $K^2 + K - (N^2 + N)/2 = 0$. Решаем квадратное уравнение и проверяем два соседних натуральных числа.

Задача E. Холодное сердце

Для решения задачи необходимо сначала выполнить перевод всех снежинок в стандартную форму: например, из 12 форм полученных в результате вращения и переворачивания можно взять форму, являющуюся лексикографически наибольшей. Затем необходимо отсортировать такие формы также лексикографически и найти количество соседних несовпадающих форм в этом упорядоченном массиве.

Задача F. Геймер Егор

Будем решать при помощи динамического программирования. Решение будет основываться на решении задаче об рюкзаке. Заведём $dp[i][j]$, — где i количество типов, которые мы обработали, j — набрали вес j , а хранится в $dp[i][j]$ максимальный суммарный рейтинг с такими параметрами. Будем обрабатывать предметы в порядке типов, в начале будем обрабатывать предметы, у которых $t_i=1$, потом два и так далее. Рассматривая очередной предмет, обновляем ответ как рюкзаке. Не забываем про восстановление ответа, для этого будем хранить ещё одну матрицу, в которой будем записывать, каким предметом был обновлен ответ.