

Задача А. Плитки

Для решения данной задачи нам нужно:

Во-первых, найти площади всех треугольников по формуле Герона и добавить их в какой-то контейнер.

Во-вторых, отсортировать наш контейнер по возрастанию площадей наших треугольников.

В-третьих, перемешаться с конца отсортированного контейнера и набирать треугольники с наибольшими площадями до тех пор, пока сумма выбранных площадей треугольников меньше искомой площади S для покрытия.

Если же мы прошлись по всему отсортированному контейнеру и сумма всех площадей треугольников все равно меньше искомой площади S для покрытия, то выводим «NO» и значение «-1», иначе выводим «YES» и минимальное количество нужное для покрытия искомой площади S .

Задача В. Кепарик и Кибер-скарабей

Для решения задачи необходимо найти длину наидлиннейшей невозрастающей подпоследовательности используя вариант динамического программирования с двоичным поиском. Состояние динамики — индекс элемента в массиве i ; результат — последнее наименьшее число подпоследовательности длины i . Поскольку данная динамика может искать только неубывающие подпоследовательности, то предварительно необходимо развернуть изначальный массив.

Задача С. Сокращение слов

Заметим, что первое правило сортирует слово по неубыванию. По завершении его применения слово, которое изначально состояло из c_0 нулей и c_1 единиц, будет содержать последовательно все c_0 нулей, за которыми следуют c_1 единиц. Например, слово 0100101001 после применения правила 1 примет вид 0000001111. Далее может получиться несколько различных ответов, в зависимости от числа нулей и единиц в слове:

1. Если число нулей и единиц нечётное, то после применения правил слово примет вид 01.
2. Если число нулей чётное, а число единиц нечётное, то после применения правил будут уничтожены все нули и чётное число единиц. Слово примет вид 1.
3. Если число нулей нечётное, а число единиц чётное, то после применения правил будут уничтожены все единицы и чётное число нулей. Слово примет вид 0.
4. Если число нулей и единиц чётное, то после применения правил будут уничтожены все нули и единицы. Ответом в этом случае будет -1 .

Задача D. Круглый магазин

Для нахождения минимального времени для покупки всех продуктов можно использовать жадный алгоритм. Создадим словарь, в котором сохраним для каждого из N товара его номер в каталоге магазина. Будем итерироваться по списку Даниила и для каждого продукта определим позицию new — его номер в каталоге. Определим минимальное расстояние между текущей позицией cur и новой позицией new . Если двигаться по часовой стрелке, то расстояние между ними будет равняться $|new - cur|$. Если двигаться против часовой стрелки, то расстояние будет равняться $N - |new - cur|$. Для каждого товара будем выбирать минимальное из этих расстояний, прибавлять его к итоговому ответу и изменять текущую позицию на новую: $cur = new$.

Задача Е. Составь больше

Пусть мы имеем строку «- 0 +». Мы поддерживаем четыре переменные: плюсы, минусы, нули и ответ. Смотрим на первый символ строки. Он самый маленький это -. Мы можем поставить вместо него 0 или +, чтобы любая перестановка была больше. Поэтому мы убираем первый символ, который заменили на + или 0 и считаем все возможные перестановки с повторениями. Так мы по очереди меняем символ и обрезаем те, которые уже прошли. Считаем сколько перестановок с повторениями можно составить обрезаая пройденные символы. В итоге их сумма будет являться ответом на вопрос.