

Задача А. Коля и сообщение

Идея задачи: Николай Гурин
Разработка: Николай Гурин, Максим Шумак
Разбор: Николай Гурин
Теги: консольный вывод

Выведите строку текста согласно условию.

Задача В. Площадь дома

Идея задачи: Николай Гурин
Разработка: Николай Гурин, Максим Шумак
Разбор: Максим Шумак
Теги: консольный ввод/вывод, площадь прямоугольника

Так как основание дома прямоугольной формы, рассчитаем площадь по формуле $a \times b$ и выведем ответ.

Задача С. Уравнение

Идея задачи: Николай Гурин
Разработка: Николай Гурин, Максим Шумак
Разбор: Максим Шумак
Теги: консольный ввод/вывод, линейные уравнения

Подставим заданные значения x и a в уравнение, найдём ответ.

Задача D. Чередование минимума и максимума

Идея задачи: Николай Гурин
Разработка: Николай Гурин
Разбор: Максим Шумак
Теги: линейные массивы, сортировка, указатели в массиве

Отсортируем массив по возрастанию и заведём два указателя l и r — для начала и конца массива — и, поочередно их сдвигая, будем выбирать элементы из начала и конца массива.

Задача Е. Коля и письмо в Альтушкино

Идея задачи: Николай Гурин
Разработка: Николай Гурин
Разбор: Максим Шумак
Теги: строковые операции, поиск слов в строке

Пройдём по строке и сделаем заглавными все символы, идущие после пробела. Также сделаем заглавным первый символ строки.

Задача Ф. Проверка сервера

Идея задачи: Максим Шумак

Разработка: Николай Гурин

Разбор: Николай Гурин

Теги: операции на строках, разбор арифметических выражений

Давайте пройдём по всем n выражениям, вычислим их значения и сравним с ответами сервера. Для этого разобьём строку на множество символов «+», «-» и на числа между ними. Так как в конце строки нет знака «+» или «-», можем добавить любой знак для унификации обработки. Если в начале строки нет «-», первым знаком будем полагать «+» — можно также добавить «+» в начало строки.

Далее будем фиксировать подстроки между знаками и переводить их в числа. В зависимости от знака перед подстрокой будем либо прибавлять, либо отнимать значение от суммы. Полученные значения выражений сравним с ответами сервера.

Задача Г. Ромашковая долина

Идея задачи: Максим Шумак

Разработка: Максим Шумак, Алексей Ропан

Разбор: Станислав Польнь

Теги: алгоритм Дейкстры, динамическое программирование по бинарной маске

Найдём кратчайшие расстояния между парами вершин с ромашками. Для этого запустим алгоритм Дейкстры из каждой из них. Далее будем считать $dp_{mask,v}$ — длина кратчайшего пути, последней вершиной которого является v , начинающийся в вершине 1 и посещающий все вершины с ромашками, которым соответствуют единица в бинарной маске $mask$. Для этого проитерировуемся в порядке возрастания $mask$ и будем фиксировать v . Обновляем результат для dp_{mask} от $2^u, u$ значением $dp_{mask,v} + dist_{u,v}$, где u — следующая вершина, которую мы посетим, а $dist_{i,j}$ — предпросчитанный массив расстояний. Тогда для любого значения $mask$ мы можем определить, существует ли путь длиной не более l — должно выполняться $dp_{mask,v} + dist_{1,v} \leq l$ (так как необходимо вернуться в вершину 1 по условию). Если это так, можем обновить ответ суммой ромашек из $mask$.

Задача Н. Петя и числа

Идея задачи: Андрей Мищенко

Разработка: Андрей Мищенко

Разбор: Андрей Мищенко

Теги: математика, бинарный поиск, жадные алгоритмы

Преобразуем заданный массив. Мы можем сделать ещё один массив b , где $b_i = f(a_i)$, здесь $f(a_i)$ — сумма цифр числа a_i . Теперь нужно выбрать k позиций так, чтобы $\frac{a_{i_1} + a_{i_2} + \dots + a_{i_k}}{b_{i_1} + b_{i_2} + \dots + b_{i_k}}$ было максимально возможным. Пусть мы предполагаем, что ответ равен c . Тогда

$$c \leq \frac{a_{i_1} + a_{i_2} + \dots + a_{i_k}}{b_{i_1} + b_{i_2} + \dots + b_{i_k}}$$

$$c \cdot (b_{i_1} + b_{i_2} + \dots + b_{i_k}) \leq a_{i_1} + a_{i_2} + \dots + a_{i_k}$$

$$0 \leq (a_{i_1} - c \cdot b_{i_1}) + (a_{i_2} - c \cdot b_{i_2}) + \dots + (a_{i_k} - c \cdot b_{i_k})$$

Мы можем перебирать значение c бинарным поиском, а позиции выбирать жадно.

Задача I. Переезд

Идея задачи: Андрей Мищенко

Разработка: Андрей Мищенко

Разбор: Андрей Мищенко

Теги: графы, реализация, компоненты двусвязности

Мы можем добавлять рёбра от большего к меньшему. Как только вершина стала лежать в каком-либо цикле, у неё появился ответ. Эту информацию можно поддерживать с помощью компонент двусвязности, как в онлайн-алгоритме поиска мостов.