

Указания к решениям задач муниципального этапа Всероссийской олимпиады школьников по информатике 2018/2019 учебного года.

7 - 8 классы.

Краснодарская краевая предметно-методическая комиссия
Всероссийской олимпиады школьников по информатике

При разборе задач настоятельно рекомендуется подробно рассматривать не только алгоритмическую составляющую решений, но и математическую, в тех случаях, когда она не тривиальна. Математический аспект решения задачи, несмотря на свою простоту, ниже разобран очень подробно с достаточно высокой степенью формализма, что иногда может препятствовать восприятию школьником. При разборе очень желательно добиться сочетания строгости и доступности изложения.

Задача 1: Поиск купе

Вычислим порядковый номер купе Миши, считая от начала поезда:

$$c_1 = [(s_1 - 1)/4] + 1 + (w_1 - 1) \cdot 9,$$

здесь $[\cdot]$ - целая часть от деления, $[(s_1 - 1)/4]$ - количество купе, предшествующих Мишиному купе в Мишином вагоне, $(w_1 - 1) \cdot 9$ - количество купе во всех вагонах, предшествующих Мишиному вагону.

Аналогично, порядковый номер купе Саши вычисляется следующим образом:

$$c_2 = [(s_2 - 1)/4] + 1 + (w_2 - 1) \cdot 9.$$

Теперь осталось вычислить абсолютное значение разности этих порядковых номеров - это и будет ответом в задаче:

$$|c_1 - c_2|.$$

Решения, построенные на итерировании по номерам мест не укладываются в отведённое время и набирают неполный балл.

Задача 2: Новое блюдо

Задача носит исключительно технический характер: по сути сам алгоритм расчёта приведён в условии, необходимо только его аккуратно реализовать. А именно, если a_i - вес i -го ингредиента, необходимый для приготовления одной порции по рецепту, b_i - вес i -го ингредиента, имеющийся в наличии у Вовы, то у Вовы его хватит на $p_i = [b_i/a_i]$ порций. Из всех таких величин необходимо выбрать минимальное, игнорируя нулевые значения (эти ингредиенты в любом случае придётся докупать в магазине), это и будет максимальное количество порций - обозначим его p . Если ненулевых значений среди p_i нет, то кладем $p = 1$. После этого для всех нулевых p_i (т.е. для всех i , таких что $b_i < a_i$) вычисляем стоимость данного ингредиента в количестве, в котором его нужно докупить в магазине, чтобы с учётом имеющегося запаса в b_i грамм приготовить p порций:

$$s_i = (p \cdot a_i - b_i) \cdot c_i.$$

Сумма s всех таких s_i и есть сумма в рублях, которую необходимо потратить в магазине. Ответом в задаче является пара чисел p и s .

Задача 3: Анонимное послание

В данной задаче нужно подсчитать для каждой буквы и каждой цифры количество её вхождений в строку-послание и количество её вхождений в строку-статью. Если разность между первой и второй величинами положительная, то это означает, что имеет место нехватка количества этой буквы или цифры в строке-статье. Сумма всех таких положительных величин (т.е. по всем буквам и цифрам, для которых разность положительна) и есть ответ в задаче. Для того, чтобы быстро считать данные количества можно использовать массив, индексруемый ASCII-кодами символов и при проходе по строке увеличивать или уменьшать соответствующее значение массива:

$$A[\text{ord}(S[i])] := A[\text{ord}(S[i])] + 1;$$

в языке Pascal. В языке Python

$$A[\text{ord}(S[i])] += 1$$

В языках C и C++:

$$A[(\text{int})S[i]]++;$$

Общее количество действий при таком подходе - порядка суммы длин строк послания и статьи.

Необходимо отметить, что в языке Pascal необходимо использовать тип AnsiString, либо тип String с директивой компилятора {\$H+}, поскольку строки обычного типа String ограничиваются длиной в 255 символов.

Задача 4: Секретное сообщение

Для решения первой подзадачи достаточно воспользоваться методом полного перебора: для каждого $i = 1, \dots, n$ рассмотреть все участки подряд идущих элементов с началом в i -ом числе и среди тех из них, которые имеют сумму, кратную k , выбрать максимальный по длине (для равных длин выбирать участок с меньшим индексом начала). Если не пересчитывать для каждого участка сумму, а прибавлять очередное число к сумме участка с тем же началом, но на единицу меньшей длины, то мы получаем решение с количеством действий $O(n^2)$, что достаточно для ограничений первой подзадачи.

Для полного решения задачи необходимо поступить следующим образом. Вычислим частичные суммы по модулю k на всех префиксах исходного массива A (индексация массива с 1):

$$S[i] = \left(\sum_{j=1}^i A[j] \right) \bmod k.$$

При последовательном вычислении частичных сумм на это у нас потребуется порядка n действий:

$$S[i+1] = (S[i] + A[i+1]) \bmod k.$$

Добавим частичную сумму $S[0] = 0$. Сумма участка от i -го до j -го элемента по модулю k может быть легко вычислена с использованием частичных сумм:

$$(A[i] + A[i+1] + \dots + A[j-1] + A[j]) \bmod k = (S[j] - S[i-1]) \bmod k.$$

Теперь задача сводится к следующей: найти две одинаковые частичные суммы с максимальной разницей индексов. Это легко сделать одновременно с их вычислением, если в массиве длины k (в силу ограничения на возможное значение k такой массив может быть создан) отмечать минимальные индексы данной частичной суммы: $m[t]$ - минимальное такое i , что $S[i] = t$ ($t = 0, \dots, k-1$, но $m[0] = 0$, поэтому нулевой элемент можно не использовать). Получаем общее количество действий $O(n)$.