

Задача А. Распределение пряности

Автор задачи и разработчик: Даниил Орешников

Первые подгруппы рассчитаны на частные решения.

- В **первой подгруппе** $n \leq 3$, то есть всего не более трех направлений; в таком случае ответ всегда не более 3, и можно просто в несколько условий перебрать все возможные исходы.
- Во **второй подгруппе** $k = 0$, то есть только одинаковые направления считаются похожими. В таком случае ответ равен максимальному количеству вхождений какого-либо числа в набор a_i : действительно, меньше нельзя, а для такого ответа легко строится пример.

В **третьей подгруппе** можно было написать полный перебор, всеми возможными способами распределяющий единицы пряности между студентами. Для такого перебора достаточно поддерживать, какие направления уже соответствуют каким студентам, и для каждого нового либо выдавать его студенту без нарушения правил, либо выдавать новому студенту. Если еще заранее упорядочить все a_i по неубыванию, такой перебор гарантированно проходил ограничения по времени.

В **четвертой подгруппе** работает решение через динамическое программирование, но мы сразу приведем решение для **четвертой и пятой подгрупп**, подводящее к полному решению. Упорядочим a_i по неубыванию и для каждого направления a_i найдем такое минимальное $j_i > i$, что $a_{j_i} - a_i > k$. Тогда все направления с i по $j_i - 1$ обязаны быть выданы разным студентам.

Утверждается, что ответ в таком случае равен $\max_{i=1}^n j_i - i$, то есть максимальной длине такого отрезка. Опять же, оценка понятна: если в каждом таком отрезке все направления достались разным студентам, то количество студентов не меньше длины максимального из отрезков. Пример же строится жадно или конструктивно. Например, если мы определили, что студентов в ответе будет m , достаточно выдать направление a_i студенту с номером $i \bmod m + 1$. Тогда любые два направления у одного студента отличаются хотя бы на $a_{i+m} - a_i$, что больше k по нашему выбору m .

В четвертой подгруппе для вычисления такого m можно было просто суммировать $\text{cnt}[x] + \text{cnt}[x + 1] + \text{cnt}[x + 2]$, перебирая все возможные x (где cnt — количество вхождений числа в последовательность a_i). В пятой же было достаточно для каждого i от 1 до n находить j_i за время $\mathcal{O}(n)$.

Полное решение же отличается только применением методов двух указателей: если $i_1 \leq i_2$, то и $j_{i_1} \leq j_{i_2}$ по логике задачи. Тогда можно двумя указателями найти j_i для каждого i в сумме за $\mathcal{O}(n)$ времени.

Задача С. Предательство Юэ

Подсчитаем для половины длины n сколько пар с суммой s и произведением p . Пусть пара кодов длины n , у которых (s, p) равно a , пар $(s - 1, p)$ равно b и пар $(s + 1, p)$ равно c а тогда в ответ надо добавить $a \cdot (a + b + c)$.

Осталось научиться считать количество таких пар. Для этого воспользуемся динамическим программированием.

Пусть у нас для длины i подсчитана пара (s, p) . Тогда на $i + 1$ мы можем поставить цифру d (от 1 до 9), и тогда для всех пар $(s + d, p * d)$ мы должны добавить ответ для пары (s, p) .

Стоит обратить внимание, что произведение и ответ может выйти за тип данных, и поэтому надо воспользоваться длинной арифметикой или написать решение на языках, в которых она встроенная, например, на языке Python.

В зависимости от эффективности реализации можно набрать различное число баллов. Но все ответы можно подсчитать у себя на компьютере и отправить программу, которая будет выводить ответы с уже подсчитанными ответами.