**Задача 1604. В Стране Дураков**

**Условие:**

Дано множество $X$ уникальных объектов (дорожных знаков), состоящее из $k$ элементов:

$$X = \{x_1, x_2, x_3, \ \ldots, x_k\}$$

Дорожный знак с номером $i$ присутствует $a_i$ раз в исходном наборе. Иными словами, можно задать отображение:

$$\varphi : X \longrightarrow \mathbb{N}$$

Но для простоты изложения лучше использовать следующее обозначение:

$$x_i \longmapsto \varphi(x_i) = a_i$$

**Предварительный анализ:**

Решением задачи считается строка длины $A = \sum_{i=1}^{k} a_i$, которая выглядит следующим образом:

$$x_{t_1}, x_{t_2}, x_{t_3}, ..., x_{t_A}$$

Здесь $t_i$ — это индекс элемента, который имеет в решении номер $i$ (имеется ввиду, что любой $i$-ый уникальный объект встречается в решении $a_i$ раз).

**Определение 1** *Смежной парой будем называть пару соседних элементов строки*

$$(x_{t_i}, x_{t_{i+1}})$$

**Определение 2** *Хорошей смежной парой будем называть смежную пару, в которой содержатся два разных элемента, т.е.*

$$(x_{t_i}, x_{t_{i+1}}) : t_i \neq t_{i+1}$$

**Определение 3** *Нехорошей смежной парой будем называть смежную пару, в которой содержатся два одинаковых элемента, т.е.*

$$(x_{t_i}, x_{t_{i+1}}) : t_i = t_{i+1}$$

**Определение 4** *Лучшим решением будем называть такое решение, в котором число хороших смежных пар принимает наибольшее возможное значение на множестве всех решений.*

**Замечание 1** *Хотя бы одно лучшее решение всегда существует*

Рассмотрим какое-нибудь лучшее решение. Предположим, что в нем встречается хотя бы одна нехорошая пара $(x_{t_i}, x_{t_{i+1}}) : t_i = t_{i+1} = c$. Тогда переставим $t_i$-ый её элемент в конец строки. У нас получится конструкция вида:

$$x_{t_1}, x_{t_2}, ..., x_{t_{i-1}}, x_{t_{i+1}}, ..., x_{t_A}, x_c$$

**Замечание 2** *В результате перестановки элемента нехорошей пары в конец строки число хороших пар не уменьшилось.*

**Замечание 3** *Последний элемент полученной строки совпадает с элементом из нехорошей пары:* $t_A = c$

**Доказательство:**

Пусть $t_A \neq c \Rightarrow$ в строке появится новая хорошая пара вида $(x_{t_A}, x_c) \Rightarrow$ число хороших пар в строке возрастет, что противоречит тому, что выбранное нами решение является лучшим решением. Следовательно, $t_A = c$.

Ч.т.д.

—

Таким образом, сколько бы раз мы не повторяли данную процедуру по поиску нехорошей пары $(x_{t_i}, x_{t_{i+1}}) : t_i = t_{i+1} = d$, мы не будем получать ничего нового. Всякий раз будет получаться, что $d = c$, т.е. единственный элемент, образующий нехорошие пары — это какой-то определенный $x_c \in X$.

Будем повторять данную процедуру до тех пор, пока левее позиции, на которой стоит $x_{t_A}$ не кончатся нехорошие пары $(x_{t_i}, x_{t_{i+1}}) : t_i = t_{i+1}$. Через какое-то конечное число шагов получим следующую конструкцию:

$$x_{t_1^*}, x_{t_2^*}, ..., x_{t_A^*}, x_c, x_c, ..., x_c$$

(Здесь имеются ввиду модифицированные индексы $t^*$)

**Лемма:**

В полученной строке $t_1^* = t_3^* = ... = t_{2i-1}^* = ... = t_A^* = c$

**Доказательство:**

Пусть $t_1^* \neq c$. Тогда переставим последний символ с конца строки перед символом $x_{t_1^*}$. У нас увеличится количество пар, что невозможно. Следовательно, $t_1^* = c$.

По построению последовательности $t_2^* \neq c$. Пусть теперь $t_3^* \neq c$. Тогда вставим между знаками $x_{t_1^*}$ и $x_{t_2^*}$ последний элемент строки. Число пар увеличится, т.к. вместо одной хорошей пары $(x_{t_1^*}, x_{t_2^*})$ появятся две хорошие пары: $(x_{t_1^*}, x_c)$ и $(x_c, x_{t_2^*})$. Это невозможно по предположению о лучшем решении, следовательно $t_3^* = c$.

Данное рассуждение можно проделать и для всех оставшихся нечетных индексов (для $t_A^*$ доказывать не нужно, т.к. $t_A^* = t_A$ по построению).

Ч.т.д.

—

Таким образом, мы получили замечательный результат. Если в нашем лучшем решении встречаются пары вида $(x_{t_i}, x_{t_{i+1}}) : t_i = t_{i+1} = c$, то в данной строке найдется всего один символ $x_c \in X$, который образует все такие пары. Причем, несложно заметить, что $a_c \geq \frac{A}{2}$.

Итак, либо наша строка имеет хотя бы одну нехорошую пару вида

$$(x_{t_i}, x_{t_{i+1}}) : t_i = t_{i+1} = c,$$

либо не имеет таковых вовсе.

**Доказательство жадности:**

Введем договоренность, что с текущего момента числа $a_i$ будут обозначать элементы массива загруженного в память программы. Они будут попрежнему обозначать количество экземпляров того или иного уникального объекта (дорожного знака), однако при каждом «выставлении» очередного дорожного знака соответствующий элемент массива будет декрементироваться.

Рассмотрим наш алгоритм. Он заполняет часть массива, заключенную в прямоугольные скобки, а затем оставшимися символами $x_c$:

$$[x_{t_1^*}, x_{t_2^*}, ..., x_{t_A^*}], x_c, x_c, ..., x_c$$

В силу своей жадности, алгоритм берет сначала те пары элементов $(l, r)$, у которых $l = x_j : a_j = max\{a_i\}, r = x_h : a_h = max\{a_i\} \setminus \{l\}$, а, следовательно, даже если в лучшем решении будет такой символ $x_c$, то мы его выберем первым, и будем выбирать и дальше в силу $a_c \geq \frac{A}{2}$.

Предположим теперь, что такого $x_c$ не существует, следовательно наш алгоритм просто заполнит часть в прямоугольных скобках. Но как он это делает?

На самом деле, если такого $x_c$ нет, то не сложно заметить, что в лучшем решении во всех смежных парах будут находиться разные элементы. Но именно так и работает наш алгоритм!

Докажем, что наш алгоритм заполняет прямоугольные скобки исключительно хорошими парами.

В самом деле, на каждом шаге мы выбираем два разных элемента. В конце концов, получается следующий результат:

$[(x_{t_1}, x_{t_2}), (x_{t_1}, x_{t_2}), ..., (x_{t_1}, x_{t_2})], [(x_{t_3}, x_{t_4}), (x_{t_3}, x_{t_4}), ..., (x_{t_3}, x_{t_4})], ...$
$[(x_{t_{n-1}}, x_{t_n}), (x_{t_{n-1}}, x_{t_n}), ..., (x_{t_{n-1}}, x_{t_n})], [x_{t_{n+1}}]$

(Последний элемент $[x_{t_{n+1}}]$ включается только при нечетном $A$)

Понятно, что элементы в круглых скобках $x_{t_{2i-1}}$ и $x_{t_{2i}}$ — разные элементы (просто по построению пар).

Сомнения могут вызывать разве что элементы, стоящие на стыке разных прямоугольных скобок, т.е. такие: $x_{t_{2i}}, x_{t_{2i+1}}$

Алгоритм работает так, что пары в круглых скобках $(x_{t_{2i-1}}, x_{t_{2i}})$ строятся так, что:

$$a_{t_{2i-1}} \geq a_{t_{2i}}$$

Цикл в прямоугольных скобках заканчивается ровно тогда, когда

$$\exists N : (a_N > a_{t_{2i-1}}) \vee (a_N > a_{t_{2i}})$$

Таким образом, после элемента $x_{t_{2i}}$ в конце цикла (в конце прямоугольных скобок) может оказаться либо элемент $x_N : a_N > a_{t_{2i-1}} \geq a_{t_{2i}}$, либо элемент $x_N : a_N > a_{t_{2i}}$. Несложно заметить, что в обоих случаях $x_N$ и $x_{t_{2i}}$ — разные элементы.

Следовательно, в полученной строке в прямоугольных скобках все пары смежных элементов разные, т.е. алгоритм выдает решение с числом хороших смежных пар равным $A - 1$, что соответствует максимально возможному числу хороших пар. Следовательно, алгоритм всегда строит только лучшие решения.

Ч.т.д.