ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА
Шумен, 11 - 13 април 2025 г.
Група D - 6 клас

Задача D?. ПОДМНОЖЕСТВА (Анализ на решението) ⏳0,3 сек. 💾 256 MB

Първа подзадача

Тази подзадача единствено изисква brute force решение - проверяваме *добротата* на всяка подредица.

Втора подзадача

Тук очевидно няма как да разгледаме всяка подредица. Затова нека разгледаме какво точно се иска от задачата - сумата от *добротите* на всички подредици. Би било хубаво, ако преформулираме условието за *доброта*. Оказва се, че *добротата* на една подредица е 1 + броя на съседните различни двойки. Тази дефиниция е много по-полезна, защото така всяка двойка елементи става независима от останалите. Затова нека "обърнем условието вместо да търсим колко допринася към отговора всяко подмножество, нека търсим колко допринася към отговора всяка двойка числа.

Нека сме фиксирали индекси $i < j$, такива че $a_i \neq a_j$. Тогава броят на подмножествата, в които тази двойка допринася, е броят на подмножествата, в които между тези два елемента на редицата няма други, тоест a_i и a_j трябва да са последователни в подредицата. Така че трябва да фиксираме и че няма включени елементи между a_i и a_j. Тогава броят на възможностите за текущия избор на i и j е $2^{i-1} \times 2^{N-j}$. Сложността е $O(N^2)$, а степените на двойката по модул могат да се изчислят предварително и да се запазят в масив.

Трета подзадача

Нека разгледаме нашето решение: за всяка двойка $i < j$, проверяваме дали $a_i \neq a_j$, и ако да - то към отговора добавяме $2^{i-1} \times 2^{N-j}$. Двата члена, които умножаваме, са независими, тоест логично е да изкараме единия извън цикъла. Така за всяко j търсим индексите $i < j$, за които $a_i \neq a_j$, сумират се 2^{i-1} и полученото се умножава по 2^{N-j}. Вместо да гледаме за елементите с различни стойности от текущия, по-умен подход би бил от общия брой дотук да извадим елементите с еднакви стойности от текущия. Това се поддържа в масив-брояч, като е нужно и компресиране на елементите (или може да се работи с map). Сложността е $O(N \log N)$.

Автор: Кирил Зулямски