

Магические ракушки

Формально, в задаче требуется максимизировать сумму двух элементов массива a , на индексы которых наложено ограничение: их побитовое «ИЛИ» не должно превосходить k .

Ключевая идея задачи в том, что разбивать пары индексов на множества, в которых $i \mid j = t$ для всех $t < k$, может быть достаточно нетривиально, а вот посчитать ответ для множеств, в которых $i \mid k \subset k$, то есть $i \subset k$ и $j \subset k$ (как множества единичных бит), уже проще. А $\bigcup_{t \leq k} \{(i, j) : i, j \subset t\} = \bigcup_{t \leq k} \{(i, j) : i \mid j = t\}$. Ведь если побитовое «ИЛИ» двух чисел меньше k , то каждое из них тоже является подмаской некоторого числа от 0 до k .

А вот для каждого $t < k$ посчитать максимальную сумму двух элементов массива, побитовое «ИЛИ» индексов которых является подмаской t , уже не очень сложно с помощью динамического программирования. Обозначим за $\text{dp}[t]$ пару максимальных элементов массива с индексами $i, j \subset t$, тогда

$$\text{dp}[t] = \text{two_max} \left(a_t, \bigcup_{b \in t} \text{dp}[t - 2^b] \right),$$

то есть объединение всех ответов для t без произвольного бита, добавить a_t , и оставить только два максимальных элемента.

Такую динамику можно посчитать за $\mathcal{O}(2^n \cdot n)$, после чего для каждого t достаточно посчитать сумму найденных двух максимумов, а по полученным величинам посчитать префиксные суммы — k -я из них и будет ответом.