
Разбор задачи «Доктор Стрэндж и выставка»

Посчитаем динамическое программирование «по битовым маскам».

$dp[i][m]$ — каким наименьшим количеством чисел от 1 до i можно получить битовую маску m , переходы следующие:

$dp[i+1][m \text{ AND } a_{i+1}] = \min(dp[i+1][m \text{ AND } a_{i+1}], dp[i][m] + 1)$ — случай, когда мы берем число a_{i+1}

$dp[i+1][m] = \min(dp[i+1][m], dp[i][m])$ — случай, когда мы не берем число a_{i+1} .

База динамического программирования:

$dp[0][2^{12} - 1] = 0$, остальные значения равны ∞ (некому числу, заведомо большему ответа, например, $n + 1$).

После чего, нужно заметить, что количество памяти, требуемое для массива dp требует $n \cdot 2^{12}$ ячеек памяти, что занимает около 320мб памяти, если использовать тип *int*.

Для того, чтобы избежать этой проблемы, нужно либо проводить все вычисления в 2-байтовом типе данных (*short int* в C++, либо *char* в Java).

Либо убрать из массива a повторяющиеся элементы, тогда n не будет превосходить 2^{12} .

Либо использовать стандартный подход: заметить, что после подсчета i -го слоя массива dp , к предыдущим слоям мы уже обращаться не будем и на подсчет dp они никак не повлияют, поэтому можно хранить только текущий слой, то есть массив $dp[0 \dots 2^{12} - 1]$, и значения динамического программирования для следующего слоя сохранять в нем же.

После подсчета динамики нужно проверить, что $dp[0] \leq k$.

Сложность этого решения составляет $n \cdot 2^{12}$.