Алгоритмы. HW#9

Тураев Тимур, 504 (SE)

1 *Найти сумму Минковского*

Пусть $A = \{a_1, a_2, \ldots, a_m\}$, $B = \{b_1, b_2, \ldots, b_k\}$ Тогда можно представить A как многочлен степени не выше $10n$: $x^{a_1} + x^{a_2} + \ldots + x^{a_m}$. Аналогично B представляем в виде $x^{b_1} + x^{b_2} + \ldots + x^{b_k}$.

Перемножив эти многочлены получим: $g_{1,1} \cdot x^{a_1+b_1} + \ldots + g_{m,k} \cdot x^{a_m+b_k}$. Достаточно очевидно, что искомое множество C есть все показатели степеней в получившемся многочлене. А заодно мы автоматически получили и ответ на вопрос: а скольими способами можно представить число $a_i + b_j$ в виде суммы двух элементов двух множеств – число способб равно коэффициенту перед соответствующей степенью, то есть $g_{i,j}$.

Так как мы умеем перемножать многочлены степени не выше $10n$ за $O(10n \cdot \log(10n)$, то искомое множество можно найти за $O(n \log n)$

2 *По заданным комплексным z_i и неотрицательным целым a_i посчитайте коэффициенты полинома $\prod_i (x-z_i)^{a_i}$. Пусть $n = \sum a_i$. Решите задачу за $O(n \log n)$*

1. Научимся сначала считать полином $(x-k)^n$ достаточно быстро.

   Если n четно, то $(x-k)^n = ((x-k)^{n/2})^2$ и можно составить рекуррентное соотношение $T(n) = T(n/2) + O(n/2 \cdot \log(n/2))$ – считаем сначала полином в степени в 2 раза меньше, потом перемножаем его с собой – так как он степени $n/2$ то это мы можем сделать за $O(n/2 \cdot \log(n/2))$. Решаем по master-method, получаем что $T(n) = O(n \cdot \log n)$

   Если n нечетно, то $(x-k)^n = ((x-k)^{n/2})^2 \cdot (x-k)$. В квадрат, как мы уже знаем, можно возвести за $O(n \log n)$. Но понятно, что зная коэффициенты $((x-k)^{n/2})^2$, умножить на $(x-k)$ достаточно просто: это почленная сумма сдвинутого вектора координат вправо на 1 плюс того же вектора, но каждый коэффициент еще умножен на $-k$. Это можно сделать за $O(n)$. Итог – те же $O(n \log n)$.

2. Давайте представим исходное произведение в виде

   $$(x-z_1)^{a_1} \cdot (x-z_2)^{a_2} \ldots (x-z_k)^{a_k}$$

   Где все a_i упорядочены по возрастанию. Тогда первое произведение мы сделаем за $O(a_2 \log a_2)$, вторoе за $O(a_3 \log a_3)$ и так далее. В итоге можно оценить это $O(a_2 \log a_2) + O(a_3 \log a_3) + \ldots + O(a_k \log a_k) = O(a_2 \log n) + O(a_3 \log n) + \ldots + O(a_k \log a_k) = O(\log n \cdot (a_2 + a_3 + \ldots + a_k)) = O(\log n \cdot (a_1 + a_2 + a_3 + \ldots + a_k)) = O(n \cdot \log n)$

3 *Про Теплицевы матрицы*

Если коротко, то Теплицева матрица это такая матрица, в которой на диагоналях стоят одинаковые числа, а значит, для ее хранения достаточно $O(n+m)$ памяти: будем хранить лишь «заголовки» диагоналей (которых $n+m-1$). То есть всю матрицу можно представить в виде вектора размера $n+m-1$: $(a_{1,m}, a_{1,m-1}, \ldots, a_{1,1}, a_{2,1}, \ldots a_{n,1})$

Научимся теперь быстро умножать матрицу на вектор $b = (b_1, b_2, \ldots, b_m)^T$. Получится вектор $c = (c_1, c_2, \ldots, c_n)^T$, такой что

$$c_i = \sum_{j=0}^{m} a_{m+i-j} \cdot b_j$$

Но этот коэффициент c_i есть ни что иное как коэффициент перед степенью $m+n-1-i$ у многочлена, являющегося результатом произведения многочленов с коэффициентами a и b, так как c_i по сути является сверткой.

Ну, а раз мы умеем перемножать многочлены за $O(n \cdot \log n)$, то произведение такой матрицы на вектора работает $O((n+m) \cdot \log(n+m))$