Дискретное преобразование Фурье

Борис Золотов (Б09)
Михаил Иванов (Б05, Б06)
Иван Казменко (Б01, Б02)
Владислав Макаров (Б03)
Арина Филимонова (Б10)

Санкт-Петербургский Государственный Университет

Четверг, 27 февраля 2025 года

Содержание

Содержание

Мотивирующая задача

Задача:

- Рассмотрим многочлены: $P(x)$ степени m и $Q(x)$ степени n.
- Как найти произведение $P(x) \cdot Q(x)$?

Мотивирующая задача

Задача:

- Рассмотрим многочлены: $P(x)$ степени m и $Q(x)$ степени n.
- Как найти произведение $P(x) \cdot Q(x)$?
- Перемножить в столбик...

Мотивирующая задача

Задача:

- Рассмотрим многочлены: $P(x)$ степени m и $Q(x)$ степени n.
- Как найти произведение $P(x) \cdot Q(x)$?
- Перемножить в столбик...
- ...за время $O(m \cdot n)$.
- А можно быстрее?

Как задать многочлен

Как можно задать многочлен?

- Коэффициенты при степенях x:
  $P(x) = a_{n-1}x^{n-1} + \ldots + a_1 x^1 + a_0 x^0$.
- Значения в n различных точках: $P(x_0), P(x_1), \ldots, P(x_{n-1})$.
  При этом можно выбрать любые удобные точки — главное, чтобы их было хотя бы n.

Как перемножить многочлены

Как перемножить два многочлена? $R(x) = P(x) \cdot Q(x)$:

- $P(x) = \sum_{i=0}^{m-1} a_i x^i$; $Q(x) = \sum_{j=0}^{n-1} b_j x^j$: $R(x) = \sum_{k=0}^{m+n-2} c_k x^k$, где

  $c_k = \sum_{i+j=k} a_i \cdot b_j$.

  Время: $O(m \cdot n)$.
- $P(x_0), P(x_1), \ldots, P(x_{m+n-2})$; $Q(x_0), Q(x_1), \ldots, Q(x_{m+n-2})$:
  $R(x_k) = P(x_k) \cdot Q(x_k)$.
  Время: $O(m + n)$.

Как умножать быстрее

Как умножить быстрее, если есть коэффициенты?

- Выбрать удобные точки.
- Перейти от коэффициентов P и Q к значениям в точках: $O(?)$.
- Перемножить значения в точках: $O(m+n)$.
- Перейти от значений R в точках к коэффициентам: $O(?)$.

Как умножать быстрее

Как умножить быстрее, если есть коэффициенты?

- Выбрать удобные точки.
- Перейти от коэффициентов P и Q к значениям в точках: $O(?)$.
- Перемножить значения в точках: $O(m+n)$.
- Перейти от значений R в точках к коэффициентам: $O(?)$.
- Переходы можно сделать за время $O((m+n)\log(m+n))$.

Как умножать быстрее

Как умножить быстрее, если есть коэффициенты?

- Выбрать удобные точки.
- Перейти от коэффициентов P и Q к значениям в точках: $O(?)$.
- Перемножить значения в точках: $O(m+n)$.
- Перейти от значений R в точках к коэффициентам: $O(?)$.
- Переходы можно сделать за время $O((m+n)\log(m+n))$.

Выбор точек — $\{\omega^i\}_{i=0}^{m+n-2}$, где ω — первообразный корень степени не меньше $m+n-1$.

Как умножать быстрее

Как умножить быстрее, если есть коэффициенты?

- Выбрать удобные точки.
- Перейти от коэффициентов P и Q к значениям в точках: $O(?)$.
- Перемножить значения в точках: $O(m+n)$.
- Перейти от значений R в точках к коэффициентам: $O(?)$.
- Переходы можно сделать за время $O((m+n)\log(m+n))$.

Переход от коэффициентов к значениям в точках — преобразование Фурье.

Как умножать быстрее

Как умножить быстрее, если есть коэффициенты?

- Выбрать удобные точки.
- Перейти от коэффициентов P и Q к значениям в точках: $O(?)$.
- Перемножить значения в точках: $O(m+n)$.
- Перейти от значений R в точках к коэффициентам: $O(?)$.
- Переходы можно сделать за время $O((m+n)\log(m+n))$.

Умножение значений в точках — покомпонентно.

Как умножать быстрее

Как умножить быстрее, если есть коэффициенты?

- Выбрать удобные точки.
- Перейти от коэффициентов P и Q к значениям в точках: $O(?)$.
- Перемножить значения в точках: $O(m+n)$.
- Перейти от значений R в точках к коэффициентам: $O(?)$.
- Переходы можно сделать за время $O((m+n)\log(m+n))$.

Переход от значений в точках к коэффициентам — обратное преобразование Фурье.

Содержание

Как выбрать точки

Как выбрать точки?

- Многочлены — скажем, над полем $\mathbb{A}$.
- Например, это может быть $\mathbb{C}$ или $\mathbb{Z}_p$ для простого p.

Как выбрать точки

Как выбрать точки?

- Многочлены — скажем, над полем $\mathbb{A}$.
- Например, это может быть $\mathbb{C}$ или $\mathbb{Z}_p$ для простого p.
- Выберем $\omega \in \mathbb{A}$ — первообразный корень из единицы степени n: $\omega^1 \neq 1$, $\omega^2 \neq 1$, …, $\omega^{n-1} \neq 1$, но $\omega^n = 1$.
- Число n должно быть больше степени любого многочлена, который нас интересует: P, Q и $R = P \cdot Q$.

Как выбрать точки

Как выбрать точки?

- Многочлены — скажем, над полем $\mathbb{A}$.
- Например, это может быть $\mathbb{C}$ или $\mathbb{Z}_p$ для простого p.
- Выберем $\omega \in \mathbb{A}$ — первообразный корень из единицы степени n: $\omega^1 \neq 1$, $\omega^2 \neq 1$, …, $\omega^{n-1} \neq 1$, но $\omega^n = 1$.
- Число n должно быть больше степени любого многочлена, который нас интересует: P, Q и $R = P \cdot Q$.
- Рассмотрим n точек $\omega^0, \omega^1, \ldots, \omega^{n-1}$.
- Будем считать значения многочленов в этих точках.

Пример для $n = 8$

Пример для $n = 8$:

$$P(x) = a_0x^0 + a_1x^1 + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7$$

Выпишем значения многочлена в точках ω^0, ω^1, …, ω^{n-1}:

$$\begin{pmatrix} P(\omega^0) \\ P(\omega^1) \\ P(\omega^2) \\ P(\omega^3) \\ P(\omega^4) \\ P(\omega^5) \\ P(\omega^6) \\ P(\omega^7) \end{pmatrix} = \begin{pmatrix} \omega^{0\cdot0} & \omega^{0\cdot1} & \omega^{0\cdot2} & \omega^{0\cdot3} & \omega^{0\cdot4} & \omega^{0\cdot5} & \omega^{0\cdot6} & \omega^{0\cdot7} \\ \omega^{1\cdot0} & \omega^{1\cdot1} & \omega^{1\cdot2} & \omega^{1\cdot3} & \omega^{1\cdot4} & \omega^{1\cdot5} & \omega^{1\cdot6} & \omega^{1\cdot7} \\ \omega^{2\cdot0} & \omega^{2\cdot1} & \omega^{2\cdot2} & \omega^{2\cdot3} & \omega^{2\cdot4} & \omega^{2\cdot5} & \omega^{2\cdot6} & \omega^{2\cdot7} \\ \omega^{3\cdot0} & \omega^{3\cdot1} & \omega^{3\cdot2} & \omega^{3\cdot3} & \omega^{3\cdot4} & \omega^{3\cdot5} & \omega^{3\cdot6} & \omega^{3\cdot7} \\ \omega^{4\cdot0} & \omega^{4\cdot1} & \omega^{4\cdot2} & \omega^{4\cdot3} & \omega^{4\cdot4} & \omega^{4\cdot5} & \omega^{4\cdot6} & \omega^{4\cdot7} \\ \omega^{5\cdot0} & \omega^{5\cdot1} & \omega^{5\cdot2} & \omega^{5\cdot3} & \omega^{5\cdot4} & \omega^{5\cdot5} & \omega^{5\cdot6} & \omega^{5\cdot7} \\ \omega^{6\cdot0} & \omega^{6\cdot1} & \omega^{6\cdot2} & \omega^{6\cdot3} & \omega^{6\cdot4} & \omega^{6\cdot5} & \omega^{6\cdot6} & \omega^{6\cdot7} \\ \omega^{7\cdot0} & \omega^{7\cdot1} & \omega^{7\cdot2} & \omega^{7\cdot3} & \omega^{7\cdot4} & \omega^{7\cdot5} & \omega^{7\cdot6} & \omega^{7\cdot7} \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix}$$

Пример для $n = 8$

Как использовать закономерности для ускорения?

$$
\begin{aligned}
p_0 &= \omega^0 a_0 + \omega^0 a_1 + \omega^0 a_2 + \omega^0 a_3 + \omega^0 a_4 + \omega^0 a_5 + \omega^0 a_6 + \omega^0 a_7 \\
p_1 &= \omega^0 a_0 + \omega^1 a_1 + \omega^2 a_2 + \omega^3 a_3 + \omega^4 a_4 + \omega^5 a_5 + \omega^6 a_6 + \omega^7 a_7 \\
p_2 &= \omega^0 a_0 + \omega^2 a_1 + \omega^4 a_2 + \omega^6 a_3 + \omega^0 a_4 + \omega^2 a_5 + \omega^4 a_6 + \omega^6 a_7 \\
p_3 &= \omega^0 a_0 + \omega^3 a_1 + \omega^6 a_2 + \omega^1 a_3 + \omega^4 a_4 + \omega^7 a_5 + \omega^2 a_6 + \omega^5 a_7 \\
p_4 &= \omega^0 a_0 + \omega^4 a_1 + \omega^0 a_2 + \omega^4 a_3 + \omega^0 a_4 + \omega^4 a_5 + \omega^0 a_6 + \omega^4 a_7 \\
p_5 &= \omega^0 a_0 + \omega^5 a_1 + \omega^2 a_2 + \omega^7 a_3 + \omega^4 a_4 + \omega^1 a_5 + \omega^6 a_6 + \omega^3 a_7 \\
p_6 &= \omega^0 a_0 + \omega^6 a_1 + \omega^4 a_2 + \omega^2 a_3 + \omega^0 a_4 + \omega^6 a_5 + \omega^4 a_6 + \omega^2 a_7 \\
p_7 &= \omega^0 a_0 + \omega^7 a_1 + \omega^6 a_2 + \omega^5 a_3 + \omega^4 a_4 + \omega^3 a_5 + \omega^2 a_6 + \omega^1 a_7
\end{aligned}
$$

$$p_i = P(\omega^i)$$

Пример для $n = 8$

Переставим столбцы: $b_i = a_{\mathrm{rev}(i)}$

$$
\begin{aligned}
p_0 &= \omega^0 a_0 + \omega^0 a_4 + \omega^0 a_2 + \omega^0 a_6 + \omega^0 a_1 + \omega^0 a_5 + \omega^0 a_3 + \omega^0 a_7 \\
p_1 &= \omega^0 a_0 + \omega^4 a_4 + \omega^2 a_2 + \omega^6 a_6 + \omega^1 a_1 + \omega^5 a_5 + \omega^3 a_3 + \omega^7 a_7 \\
p_2 &= \omega^0 a_0 + \omega^0 a_4 + \omega^4 a_2 + \omega^4 a_6 + \omega^2 a_1 + \omega^2 a_5 + \omega^6 a_3 + \omega^6 a_7 \\
p_3 &= \omega^0 a_0 + \omega^4 a_4 + \omega^6 a_2 + \omega^2 a_6 + \omega^3 a_1 + \omega^7 a_5 + \omega^1 a_3 + \omega^5 a_7 \\
p_4 &= \omega^0 a_0 + \omega^0 a_4 + \omega^0 a_2 + \omega^0 a_6 + \omega^4 a_1 + \omega^4 a_5 + \omega^4 a_3 + \omega^4 a_7 \\
p_5 &= \omega^0 a_0 + \omega^4 a_4 + \omega^2 a_2 + \omega^6 a_6 + \omega^5 a_1 + \omega^1 a_5 + \omega^7 a_3 + \omega^3 a_7 \\
p_6 &= \omega^0 a_0 + \omega^0 a_4 + \omega^4 a_2 + \omega^4 a_6 + \omega^6 a_1 + \omega^6 a_5 + \omega^2 a_3 + \omega^2 a_7 \\
p_7 &= \omega^0 a_0 + \omega^4 a_4 + \omega^6 a_2 + \omega^2 a_6 + \omega^7 a_1 + \omega^3 a_5 + \omega^5 a_3 + \omega^1 a_7
\end{aligned}
$$

$$
\begin{aligned}
b_{000} &= a_{000} & b_{001} &= a_{100} & b_{010} &= a_{010} & b_{011} &= a_{110} \\
b_{100} &= a_{001} & b_{101} &= a_{101} & b_{110} &= a_{011} & b_{111} &= a_{111}
\end{aligned}
$$

Пример для $n = 8$

Переставим столбцы: $b_i = a_{\text{rev}(i)}$

$$
\begin{aligned}
p_0 &= \omega^0 b_0 + \omega^0 b_1 + \omega^0 b_2 + \omega^0 b_3 + \omega^0 b_4 + \omega^0 b_5 + \omega^0 b_6 + \omega^0 b_7 \\
p_1 &= \omega^0 b_0 + \omega^4 b_1 + \omega^2 b_2 + \omega^6 b_3 + \omega^1 b_4 + \omega^5 b_5 + \omega^3 b_6 + \omega^7 b_7 \\
p_2 &= \omega^0 b_0 + \omega^0 b_1 + \omega^4 b_2 + \omega^4 b_3 + \omega^2 b_4 + \omega^2 b_5 + \omega^6 b_6 + \omega^6 b_7 \\
p_3 &= \omega^0 b_0 + \omega^4 b_1 + \omega^6 b_2 + \omega^2 b_3 + \omega^3 b_4 + \omega^7 b_5 + \omega^1 b_6 + \omega^5 b_7 \\
p_4 &= \omega^0 b_0 + \omega^0 b_1 + \omega^0 b_2 + \omega^0 b_3 + \omega^4 b_4 + \omega^4 b_5 + \omega^4 b_6 + \omega^4 b_7 \\
p_5 &= \omega^0 b_0 + \omega^4 b_1 + \omega^2 b_2 + \omega^6 b_3 + \omega^5 b_4 + \omega^1 b_5 + \omega^7 b_6 + \omega^3 b_7 \\
p_6 &= \omega^0 b_0 + \omega^0 b_1 + \omega^4 b_2 + \omega^4 b_3 + \omega^6 b_4 + \omega^6 b_5 + \omega^2 b_6 + \omega^2 b_7 \\
p_7 &= \omega^0 b_0 + \omega^4 b_1 + \omega^6 b_2 + \omega^2 b_3 + \omega^7 b_4 + \omega^3 b_5 + \omega^5 b_6 + \omega^1 b_7
\end{aligned}
$$

$$
\begin{aligned}
b_0 &= a_0 \quad b_1 = a_4 \quad b_2 = a_2 \quad b_3 = a_6 \\
b_4 &= a_1 \quad b_5 = a_5 \quad b_6 = a_3 \quad b_7 = a_7
\end{aligned}
$$

Пример для $n = 8$

Группируем пары соседей! И помним, что $\omega^8 = \omega^0 = 1$.

$$
\begin{aligned}
p_0 &= \omega^0(b_0 + \omega^0 b_1) + \omega^0(b_2 + \omega^0 b_3) + \omega^0(b_4 + \omega^0 b_5) + \omega^0(b_6 + \omega^0 b_7) \\
p_1 &= \omega^0(b_0 + \omega^4 b_1) + \omega^2(b_2 + \omega^4 b_3) + \omega^1(b_4 + \omega^4 b_5) + \omega^3(b_6 + \omega^4 b_7) \\
p_2 &= \omega^0(b_0 + \omega^0 b_1) + \omega^4(b_2 + \omega^0 b_3) + \omega^2(b_4 + \omega^0 b_5) + \omega^6(b_6 + \omega^0 b_7) \\
p_3 &= \omega^0(b_0 + \omega^4 b_1) + \omega^6(b_2 + \omega^4 b_3) + \omega^3(b_4 + \omega^4 b_5) + \omega^1(b_6 + \omega^4 b_7) \\
p_4 &= \omega^0(b_0 + \omega^0 b_1) + \omega^0(b_2 + \omega^0 b_3) + \omega^4(b_4 + \omega^0 b_5) + \omega^4(b_6 + \omega^0 b_7) \\
p_5 &= \omega^0(b_0 + \omega^4 b_1) + \omega^2(b_2 + \omega^4 b_3) + \omega^5(b_4 + \omega^4 b_5) + \omega^7(b_6 + \omega^4 b_7) \\
p_6 &= \omega^0(b_0 + \omega^0 b_1) + \omega^4(b_2 + \omega^0 b_3) + \omega^6(b_4 + \omega^0 b_5) + \omega^2(b_6 + \omega^0 b_7) \\
p_7 &= \omega^0(b_0 + \omega^4 b_1) + \omega^6(b_2 + \omega^4 b_3) + \omega^7(b_4 + \omega^4 b_5) + \omega^5(b_6 + \omega^4 b_7)
\end{aligned}
$$

$$
\begin{aligned}
c_0 &= b_0 + \omega^0 b_1 \quad & c_1 &= b_0 + \omega^4 b_1 \quad & c_2 &= b_2 + \omega^0 b_3 \quad & c_3 &= b_2 + \omega^4 b_3 \\
c_4 &= b_4 + \omega^0 b_5 \quad & c_5 &= b_4 + \omega^4 b_5 \quad & c_6 &= b_6 + \omega^0 b_7 \quad & c_7 &= b_6 + \omega^4 b_7
\end{aligned}
$$

Пример для $n = 8$

$$
\begin{aligned}
p_0 &= \omega^0 c_0 + \omega^0 c_2 + \omega^0 c_4 + \omega^0 c_6 \\
p_1 &= \omega^0 c_1 + \omega^2 c_3 + \omega^1 c_5 + \omega^3 c_7 \\
p_2 &= \omega^0 c_0 + \omega^4 c_2 + \omega^2 c_4 + \omega^6 c_6 \\
p_3 &= \omega^0 c_1 + \omega^6 c_3 + \omega^3 c_5 + \omega^1 c_7 \\
p_4 &= \omega^0 c_0 + \omega^0 c_2 + \omega^4 c_4 + \omega^4 c_6 \\
p_5 &= \omega^0 c_1 + \omega^2 c_3 + \omega^5 c_5 + \omega^7 c_7 \\
p_6 &= \omega^0 c_0 + \omega^4 c_2 + \omega^6 c_4 + \omega^2 c_6 \\
p_7 &= \omega^0 c_1 + \omega^6 c_3 + \omega^7 c_5 + \omega^5 c_7
\end{aligned}
$$

Пример для $n = 8$

Продолжаем группировать соседние элементы...

$$
\begin{aligned}
p_0 &= \omega^0(c_0 + \omega^0 c_2) + \omega^0(c_4 + \omega^0 c_6) \\
p_1 &= \omega^0(c_1 + \omega^2 c_3) + \omega^1(c_5 + \omega^2 c_7) \\
p_2 &= \omega^0(c_0 + \omega^4 c_2) + \omega^2(c_4 + \omega^4 c_6) \\
p_3 &= \omega^0(c_1 + \omega^6 c_3) + \omega^3(c_5 + \omega^6 c_7) \\
p_4 &= \omega^0(c_0 + \omega^0 c_2) + \omega^4(c_4 + \omega^0 c_6) \\
p_5 &= \omega^0(c_1 + \omega^2 c_3) + \omega^5(c_5 + \omega^2 c_7) \\
p_6 &= \omega^0(c_0 + \omega^4 c_2) + \omega^6(c_4 + \omega^4 c_6) \\
p_7 &= \omega^0(c_1 + \omega^6 c_3) + \omega^7(c_5 + \omega^6 c_7)
\end{aligned}
$$

$$
\begin{aligned}
d_0 &= c_0 + \omega^0 c_2 \quad & d_1 &= c_1 + \omega^2 c_3 \quad & d_2 &= c_0 + \omega^4 c_2 \quad & d_3 &= c_1 + \omega^6 c_3 \\
d_4 &= c_4 + \omega^0 c_6 \quad & d_5 &= c_5 + \omega^2 c_7 \quad & d_6 &= c_4 + \omega^4 c_6 \quad & d_7 &= c_5 + \omega^6 c_7
\end{aligned}
$$

Пример для $n = 8$

$$
\begin{aligned}
p_0 &= \omega^0 d_0 + \omega^0 d_4 \\
p_1 &= \omega^0 d_1 + \omega^1 d_5 \\
p_2 &= \omega^0 d_2 + \omega^2 d_6 \\
p_3 &= \omega^0 d_3 + \omega^3 d_7 \\
p_4 &= \omega^0 d_0 + \omega^4 d_4 \\
p_5 &= \omega^0 d_1 + \omega^5 d_5 \\
p_6 &= \omega^0 d_2 + \omega^6 d_6 \\
p_7 &= \omega^0 d_3 + \omega^7 d_7
\end{aligned}
$$

Пример для $n = 8$

И ещё раз...

$$
\begin{aligned}
p_0 &= \omega^0(d_0 + \omega^0 d_4) \\
p_1 &= \omega^0(d_1 + \omega^1 d_5) \\
p_2 &= \omega^0(d_2 + \omega^2 d_6) \\
p_3 &= \omega^0(d_3 + \omega^3 d_7) \\
p_4 &= \omega^0(d_0 + \omega^4 d_4) \\
p_5 &= \omega^0(d_1 + \omega^5 d_5) \\
p_6 &= \omega^0(d_2 + \omega^6 d_6) \\
p_7 &= \omega^0(d_3 + \omega^7 d_7)
\end{aligned}
$$

$$
\begin{aligned}
e_0 &= d_0 + \omega^0 d_4 \quad & e_1 &= d_1 + \omega^1 d_5 \quad & e_2 &= d_2 + \omega^2 d_6 \quad & e_3 &= d_3 + \omega^3 d_7 \\
e_4 &= d_0 + \omega^4 d_4 \quad & e_5 &= d_1 + \omega^5 d_5 \quad & e_6 &= d_2 + \omega^6 d_6 \quad & e_7 &= d_3 + \omega^7 d_7
\end{aligned}
$$

Пример для $n = 8$

$$
\begin{aligned}
p_0 &= \omega^0 e_0 \\
p_1 &= \omega^0 e_1 \\
p_2 &= \omega^0 e_2 \\
p_3 &= \omega^0 e_3 \\
p_4 &= \omega^0 e_4 \\
p_5 &= \omega^0 e_5 \\
p_6 &= \omega^0 e_6 \\
p_7 &= \omega^0 e_7
\end{aligned}
$$

Пример для $n = 8$

Получилось $\log_2 n$ шагов.

$$
\begin{aligned}
p_0 &= e_0 \\
p_1 &= e_1 \\
p_2 &= e_2 \\
p_3 &= e_3 \\
p_4 &= e_4 \\
p_5 &= e_5 \\
p_6 &= e_6 \\
p_7 &= e_7
\end{aligned}
$$

Код

```
1   int const logLimit = 3;
2   int const limit = 1 << logLimit;
3
4   vector <int> rev;
5
6   void calcRev () {
7       rev = vector <int> (limit, 0);
8       for (int i = 0; i < limit; i++)
9           for (int k = 0; k < logLimit; k++)
10              if (i & (1 << k))
11                  rev[i] ^= 1 << (logLimit - k - 1);
12  }
13
14  using Num = complex <double>;
15
16  double const Pi = acos (-1.0);
17
18  vector <Num> z;
19
20  void calcZ () {
21      z = vector <Num> (limit);
22      for (int i = 0; i < limit; i++)
23          z[i] = Num (cos (i * 2 * Pi / limit),
24                      sin (i * 2 * Pi / limit));
25  }
```

```
rev[0] = 0
rev[1] = 4
rev[2] = 2
rev[3] = 6
rev[4] = 1
rev[5] = 5
rev[6] = 3
rev[7] = 7
```

```
z[0] = (1,0)
z[1] = (0.707107,0.707107)
z[2] = (6.12303e-017,1)
z[3] = (-0.707107,0.707107)
z[4] = (-1,1.22461e-016)
z[5] = (-0.707107,-0.707107)
z[6] = (-1.83691e-016,-1)
z[7] = (0.707107,-0.707107)
```

Код

```cpp
#include <cmath>
#include <complex>
#include <iostream>
#include <vector>
using namespace std;

// ...

int main () {
    calcRev ();
    calcZ ();
    vector <Num> a (limit, Num (0));
    int m;
    cin >> m;
    for (int i = 0; i < m; i++) {
        int x;
        cin >> x;
        a[i] = Num (x);
    }
    auto res = fft (a);
    for (int i = 0; i < limit; i++)
        cout << res[i] << endl;
    return 0;
}
```

Ввод:

```
4
1 0 3 2
```

Вывод:

```
(6,0)
(-0.414214,4.41421)
(-2,-2)
(2.41421,-1.58579)
(2,2.44921e-016)
(2.41421,1.58579)
(-2,2)
(-0.414214,-4.41421)
```

Код

```
1   vector <Num> fft (const vector <Num> & a0) {
2       vector <Num> a = a0;
3       for (int i = 0; i < limit; i++)
4           if (i < rev[i])
5               swap (a[i], a[rev[i]]);
6       for (int k = 0, span = 1, step = limit / 2;
7         k < logLimit; k++, span *= 2, step /= 2) {
8           for (int i = 0; i < limit; i += 2 * span)
9               for (int j = 0; j < span; j++) {
10                  int u = i + j;
11                  int v = i + j + span;
12                  Num x = a[u] + a[v] *
13                    z[j * step];
14                  Num y = a[u] + a[v] *
15                    z[j * step + limit / 2];
16                  a[u] = x;
17                  a[v] = y;
18              }
19      }
20      return a;
21  }
```

Ввод:

```
4
1 0 3 2
```

Вывод:

```
(6,0)
(-0.414214,4.41421)
(-2,-2)
(2.41421,-1.58579)
(2,2.44921e-016)
(2.41421,1.58579)
(-2,2)
(-0.414214,-4.41421)
```

```
k = 0, span = 1, step = 4
a[0] = a[0] + a[1] * z[0]
a[1] = a[0] + a[1] * z[4]
a[2] = a[2] + a[3] * z[0]
a[3] = a[2] + a[3] * z[4]
a[4] = a[4] + a[5] * z[0]
a[5] = a[4] + a[5] * z[4]
a[6] = a[6] + a[7] * z[0]
a[7] = a[6] + a[7] * z[4]
```

Код

```
1   vector <Num> fft (const vector <Num> & a0) {
2       vector <Num> a = a0;
3       for (int i = 0; i < limit; i++)
4           if (i < rev[i])
5               swap (a[i], a[rev[i]]);
6       for (int k = 0, span = 1, step = limit / 2;
7         k < logLimit; k++, span *= 2, step /= 2) {
8           for (int i = 0; i < limit; i += 2 * span)
9               for (int j = 0; j < span; j++) {
10                  int u = i + j;
11                  int v = i + j + span;
12                  Num x = a[u] + a[v] *
13                    z[j * step];
14                  Num y = a[u] + a[v] *
15                    z[j * step + limit / 2];
16                  a[u] = x;
17                  a[v] = y;
18              }
19      }
20      return a;
21  }
```

Ввод:

```
4
1 0 3 2
```

Вывод:

```
(6,0)
(-0.414214,4.41421)
(-2,-2)
(2.41421,-1.58579)
(2,2.44921e-016)
(2.41421,1.58579)
(-2,2)
(-0.414214,-4.41421)
```

```
k = 1, span = 2, step = 2
a[0] = a[0] + a[2] * z[0]
a[2] = a[0] + a[2] * z[4]
a[1] = a[1] + a[3] * z[2]
a[3] = a[1] + a[3] * z[6]
a[4] = a[4] + a[6] * z[0]
a[6] = a[4] + a[6] * z[4]
a[5] = a[5] + a[7] * z[2]
a[7] = a[5] + a[7] * z[6]
```

Код

```
1   vector <Num> fft (const vector <Num> & a0) {
2       vector <Num> a = a0;
3       for (int i = 0; i < limit; i++)
4           if (i < rev[i])
5               swap (a[i], a[rev[i]]);
6       for (int k = 0, span = 1, step = limit / 2;
7         k < logLimit; k++, span *= 2, step /= 2) {
8           for (int i = 0; i < limit; i += 2 * span)
9               for (int j = 0; j < span; j++) {
10                  int u = i + j;
11                  int v = i + j + span;
12                  Num x = a[u] + a[v] *
13                    z[j * step];
14                  Num y = a[u] + a[v] *
15                    z[j * step + limit / 2];
16                  a[u] = x;
17                  a[v] = y;
18              }
19      }
20      return a;
21  }
```

Ввод:

```
4
1 0 3 2
```

Вывод:

```
(6,0)
(-0.414214,4.41421)
(-2,-2)
(2.41421,-1.58579)
(2,2.44921e-016)
(2.41421,1.58579)
(-2,2)
(-0.414214,-4.41421)
```

```
k = 2, span = 4, step = 1
a[0] = a[0] + a[4] * z[0]
a[4] = a[0] + a[4] * z[4]
a[1] = a[1] + a[5] * z[1]
a[5] = a[1] + a[5] * z[5]
a[2] = a[2] + a[6] * z[2]
a[6] = a[2] + a[6] * z[6]
a[3] = a[3] + a[7] * z[3]
a[7] = a[3] + a[7] * z[7]
```

Код

```
1  vector <Num> fftSlow (const vector <Num> & a0) {
2      vector <Num> res (limit, Num (0));
3      for (int i = 0; i < limit; i++)
4          for (int j = 0; j < int (a0.size ()); j++)
5              res[i] += a0[j] * z[(j * i) % limit];
6      return res;
7  }
```

Ввод:

```
4
1 0 3 2
```

Вывод:

```
(6,0)
(-0.414214,4.41421)
(-2,-2)
(2.41421,-1.58579)
(2,2.44921e-016)
(2.41421,1.58579)
(-2,2)
(-0.414214,-4.41421)
```

Обратное преобразование

Как сделать обратное преобразование?

Обратное преобразование

Как сделать обратное преобразование?
Умножить на обратную матрицу.

Обратное преобразование

Как сделать обратное преобразование?
Умножить на обратную матрицу.

$$\begin{pmatrix} P(\omega^0) \\ P(\omega^1) \\ P(\omega^2) \\ P(\omega^3) \end{pmatrix} = \begin{pmatrix} \omega^{0\cdot 0} & \omega^{0\cdot 1} & \omega^{0\cdot 2} & \omega^{0\cdot 3} \\ \omega^{1\cdot 0} & \omega^{1\cdot 1} & \omega^{1\cdot 2} & \omega^{1\cdot 3} \\ \omega^{2\cdot 0} & \omega^{2\cdot 1} & \omega^{2\cdot 2} & \omega^{2\cdot 3} \\ \omega^{3\cdot 0} & \omega^{3\cdot 1} & \omega^{3\cdot 2} & \omega^{3\cdot 3} \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

Обратное преобразование

Как сделать обратное преобразование?
Умножить на обратную матрицу.

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} \omega^{0\cdot 0} & \omega^{0\cdot 1} & \omega^{0\cdot 2} & \omega^{0\cdot 3} \\ \omega^{1\cdot 0} & \omega^{1\cdot 1} & \omega^{1\cdot 2} & \omega^{1\cdot 3} \\ \omega^{2\cdot 0} & \omega^{2\cdot 1} & \omega^{2\cdot 2} & \omega^{2\cdot 3} \\ \omega^{3\cdot 0} & \omega^{3\cdot 1} & \omega^{3\cdot 2} & \omega^{3\cdot 3} \end{pmatrix}^{-1} \cdot \begin{pmatrix} P(\omega^0) \\ P(\omega^1) \\ P(\omega^2) \\ P(\omega^3) \end{pmatrix}$$

Обратное преобразование

Как сделать обратное преобразование?

Умножить на обратную матрицу.

Заметим, что:

$$\begin{pmatrix} \omega^{0\cdot 0} & \omega^{0\cdot 1} & \omega^{0\cdot 2} & \omega^{0\cdot 3} \\ \omega^{1\cdot 0} & \omega^{1\cdot 1} & \omega^{1\cdot 2} & \omega^{1\cdot 3} \\ \omega^{2\cdot 0} & \omega^{2\cdot 1} & \omega^{2\cdot 2} & \omega^{2\cdot 3} \\ \omega^{3\cdot 0} & \omega^{3\cdot 1} & \omega^{3\cdot 2} & \omega^{3\cdot 3} \end{pmatrix} \cdot \begin{pmatrix} \omega^{-0\cdot 0} & \omega^{-0\cdot 1} & \omega^{-0\cdot 2} & \omega^{-0\cdot 3} \\ \omega^{-1\cdot 0} & \omega^{-1\cdot 1} & \omega^{-1\cdot 2} & \omega^{-1\cdot 3} \\ \omega^{-2\cdot 0} & \omega^{-2\cdot 1} & \omega^{-2\cdot 2} & \omega^{-2\cdot 3} \\ \omega^{-3\cdot 0} & \omega^{-3\cdot 1} & \omega^{-3\cdot 2} & \omega^{-3\cdot 3} \end{pmatrix} =$$

Обратное преобразование

Как сделать обратное преобразование?

Умножить на обратную матрицу.

Заметим, что:

$$\begin{pmatrix} \omega^{0\cdot0} & \omega^{0\cdot1} & \omega^{0\cdot2} & \omega^{0\cdot3} \\ \omega^{1\cdot0} & \omega^{1\cdot1} & \omega^{1\cdot2} & \omega^{1\cdot3} \\ \omega^{2\cdot0} & \omega^{2\cdot1} & \omega^{2\cdot2} & \omega^{2\cdot3} \\ \omega^{3\cdot0} & \omega^{3\cdot1} & \omega^{3\cdot2} & \omega^{3\cdot3} \end{pmatrix} \cdot \begin{pmatrix} \omega^{-0\cdot0} & \omega^{-0\cdot1} & \omega^{-0\cdot2} & \omega^{-0\cdot3} \\ \omega^{-1\cdot0} & \omega^{-1\cdot1} & \omega^{-1\cdot2} & \omega^{-1\cdot3} \\ \omega^{-2\cdot0} & \omega^{-2\cdot1} & \omega^{-2\cdot2} & \omega^{-2\cdot3} \\ \omega^{-3\cdot0} & \omega^{-3\cdot1} & \omega^{-3\cdot2} & \omega^{-3\cdot3} \end{pmatrix} =$$

$$= \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n & 0 \\ 0 & 0 & 0 & n \end{pmatrix}$$

Обратное преобразование

Как сделать обратное преобразование?
Умножить на обратную матрицу.
Поэтому:

$$\begin{pmatrix} \omega^{0\cdot 0} & \omega^{0\cdot 1} & \omega^{0\cdot 2} & \omega^{0\cdot 3} \\ \omega^{1\cdot 0} & \omega^{1\cdot 1} & \omega^{1\cdot 2} & \omega^{1\cdot 3} \\ \omega^{2\cdot 0} & \omega^{2\cdot 1} & \omega^{2\cdot 2} & \omega^{2\cdot 3} \\ \omega^{3\cdot 0} & \omega^{3\cdot 1} & \omega^{3\cdot 2} & \omega^{3\cdot 3} \end{pmatrix}^{-1} = \frac{1}{n} \cdot \begin{pmatrix} \omega^{-0\cdot 0} & \omega^{-0\cdot 1} & \omega^{-0\cdot 2} & \omega^{-0\cdot 3} \\ \omega^{-1\cdot 0} & \omega^{-1\cdot 1} & \omega^{-1\cdot 2} & \omega^{-1\cdot 3} \\ \omega^{-2\cdot 0} & \omega^{-2\cdot 1} & \omega^{-2\cdot 2} & \omega^{-2\cdot 3} \\ \omega^{-3\cdot 0} & \omega^{-3\cdot 1} & \omega^{-3\cdot 2} & \omega^{-3\cdot 3} \end{pmatrix}$$

Обратное преобразование

Как сделать обратное преобразование?
Умножить на обратную матрицу.
Поэтому:

$$\begin{pmatrix} \omega^{0\cdot0} & \omega^{0\cdot1} & \omega^{0\cdot2} & \omega^{0\cdot3} \\ \omega^{1\cdot0} & \omega^{1\cdot1} & \omega^{1\cdot2} & \omega^{1\cdot3} \\ \omega^{2\cdot0} & \omega^{2\cdot1} & \omega^{2\cdot2} & \omega^{2\cdot3} \\ \omega^{3\cdot0} & \omega^{3\cdot1} & \omega^{3\cdot2} & \omega^{3\cdot3} \end{pmatrix}^{-1} = \frac{1}{n} \cdot \begin{pmatrix} \omega^{-0\cdot0} & \omega^{-0\cdot1} & \omega^{-0\cdot2} & \omega^{-0\cdot3} \\ \omega^{-1\cdot0} & \omega^{-1\cdot1} & \omega^{-1\cdot2} & \omega^{-1\cdot3} \\ \omega^{-2\cdot0} & \omega^{-2\cdot1} & \omega^{-2\cdot2} & \omega^{-2\cdot3} \\ \omega^{-3\cdot0} & \omega^{-3\cdot1} & \omega^{-3\cdot2} & \omega^{-3\cdot3} \end{pmatrix}$$

Заметим ещё, что $\omega^{-k} = \omega^{n-k}$.

Код обратного преобразования

```
1   vector <Num> fft (const vector <Num> & a0, bool inv = false) {
2       vector <Num> a = a0;
3       for (int i = 0; i < limit; i++)
4           if (i < rev[i])
5               swap (a[i], a[rev[i]]);
6       if (inv)
7           reverse (z.begin () + 1, z.end ());
8       for (int k = 0, span = 1, step = limit / 2; k < logLimit;
9         k++, span *= 2, step /= 2) {
10          for (int i = 0; i < limit; i += 2 * span)
11              for (int j = 0; j < span; j++) {
12                  int u = i + j;
13                  int v = i + j + span;
14                  Num x = a[u] + a[v] * z[j * step];
15                  Num y = a[u] + a[v] * z[j * step + limit / 2];
16                  a[u] = x;
17                  a[v] = y;
18              }
19      }
20      if (inv) {
21          reverse (z.begin () + 1, z.end ());
22          for (int i = 0; i < limit; i++)
23              a[i] /= limit;
24      }
25      return a;
26  }
```

Ввод:

```
4
1 0 3 2
```

Вывод:

```
(1,2.5266e-016)
(-2.35514e-016,2.58379e-016)
(3,-6.66974e-016)
(2,-6.96749e-016)
(1.66533e-016,9.18455e-017)
(2.35514e-016,2.58379e-016)
(0,2.00008e-016)
(0,1.91429e-016)
```

Содержание

Умножение чисел

Заданы два числа. Найдите их произведение.

Умножение чисел

Заданы два числа. Найдите их произведение.

- Будем считать, что число — это многочлен, его коэффициенты — цифры.
- Например, $123 = 1 \cdot x^2 + 2 \cdot x^1 + 3 \cdot x^0$.

Умножение чисел

Заданы два числа. Найдите их произведение.

- Будем считать, что число — это многочлен, его коэффициенты — цифры.
- Например, $123 = 1 \cdot x^2 + 2 \cdot x^1 + 3 \cdot x^0$.
- Найдём произведение многочленов:
  $987 \cdot 123 = (9x^2 + 8x + 7) \cdot (1x^2 + 2x + 3)$.

Умножение чисел

Заданы два числа. Найдите их произведение.

- Будем считать, что число — это многочлен, его коэффициенты — цифры.
- Например, $123 = 1 \cdot x^2 + 2 \cdot x^1 + 3 \cdot x^0$.
- Найдём произведение многочленов:
  $987 \cdot 123 = (9x^2 + 8x + 7) \cdot (1x^2 + 2x + 3)$.
- Получится
  $(9 \cdot 1)x^4 + (9 \cdot 2 + 8 \cdot 1)x^3 + (9 \cdot 3 + 8 \cdot 2 + 7 \cdot 1)x^2 + (8 \cdot 3 + 7 \cdot 2)x + (7 \cdot 3)$.
- Это равно $9x^4 + 26x^3 + 50x^2 + 38x + 21$.

Умножение чисел

Заданы два числа. Найдите их произведение.

- Будем считать, что число — это многочлен, его коэффициенты — цифры.
- Например, $123 = 1 \cdot x^2 + 2 \cdot x^1 + 3 \cdot x^0$.
- Найдём произведение многочленов:
  $987 \cdot 123 = (9x^2 + 8x + 7) \cdot (1x^2 + 2x + 3)$.
- Получится
  $(9 \cdot 1)x^4 + (9 \cdot 2 + 8 \cdot 1)x^3 + (9 \cdot 3 + 8 \cdot 2 + 7 \cdot 1)x^2 + (8 \cdot 3 + 7 \cdot 2)x + (7 \cdot 3)$.
- Это равно $9x^4 + 26x^3 + 50x^2 + 38x + 21$.
- Теперь пройдём от младших цифр к старшим и сделаем переносы:
- $21 + 0 = 21 \to 1;\ 38 + 2 = 40 \to 0;\ 50 + 4 = 54 \to 4;$
  $26 + 5 = 31 \to 1;\ 9 + 3 = 12 \to 2;\ 0 + 1 = 1 \to 1.$
- Ответ: 121 401.

Код

```
1   vector <Num> readNumber ()
2   {
3       string s;
4       cin >> s;
5       vector <Num> res (limit, Num (0));
6       int n = int (s.size ());
7       for (int i = 0; i < n; i++)
8           res[i] = Num (s[n - 1 - i] - '0');
9       return res;
10  }
```

Ввод:

```
987
123
```

Вывод:

```
121401
```

Код

```
1       auto a = readNumber ();
2       auto b = readNumber ();
3
4       auto fa = fft (a);
5       auto fb = fft (b);
6       auto fc = vector <Num> (limit);
7       for (int i = 0; i < limit; i++)
8           fc[i] = fa[i] * fb[i];
9       auto c = fft (fc, true);
10
11      vector <int> res (limit);
12      long long carry = 0;
13      for (int i = 0; i < limit; i++) {
14          carry += (long long) (c[i].real () + 0.5);
15          res[i] = carry % 10;
16          carry /= 10;
17      }
18
19      bool started = false;
20      for (int i = limit - 1; i >= 0; i--) {
21          started |= (i == 0) | (res[i] != 0);
22          if (started)
23              cout << res[i];
24      }
25      cout << endl;
```

Ввод:

```
987
123
```

Вывод:

```
121401
```

Циклические строки

Есть две циклические строки длины $n = 2^k$ из нулей и единиц. Они записываются одна под другой, и единицы сверху и снизу совмещаются. Как их повернуть, чтобы совместить как можно больше единиц?

Циклические строки

Есть две циклические строки длины $n = 2^k$ из нулей и единиц. Они записываются одна под другой, и единицы сверху и снизу совмещаются. Как их повернуть, чтобы совместить как можно больше единиц?

Пример: $s = 10110010$, $t = 01010111$. Зафиксируем s и будем циклически двигать t вправо:

$s^0 = 10110010$
$t^0 = 01010111$
$t^1 = 10101011$
$t^2 = 11010101$
$t^3 = 11101010$
$t^4 = 01110101$
$t^5 = 10111010$
$t^6 = 01011101$
$t^7 = 10101110$

Циклические строки

Есть две циклические строки длины $n = 2^k$ из нулей и единиц. Они записываются одна под другой, и единицы сверху и снизу совмещаются. Как их повернуть, чтобы совместить как можно больше единиц?

Пример: $s = 10110010$, $t = 01010111$. Зафиксируем s и будем циклически двигать t вправо:

	Ответ:
$s^0 = 10110010$	
$t^0 = 01010111$	2
$t^1 = 10101011$	3
$t^2 = 11010101$	2
$t^3 = 11101010$	3
$t^4 = 01110101$	2
$t^5 = 10111010$	4
$t^6 = 01011101$	1
$t^7 = 10101110$	3

Циклические строки

Есть две циклические строки длины $n = 2^k$ из нулей и единиц. Они записываются одна под другой, и единицы сверху и снизу совмещаются. Как их повернуть, чтобы совместить как можно больше единиц?

Ответ: если сдвинуть t на 5 позиций вправо, получится совместить 4 единицы.

Эта проверка занимает $O(n^2)$ времени — можно ли быстрее?

Решение

Решение:

- Представим s и t как многочлены с коэффициентами 0 и 1.
- Хочется сделать так, чтобы коэффициент у произведения многочленов был равен количеству совпадающих единиц.
- Пусть мы сдвинули t на k позиций вправо. Тогда ответ равен $s_0 t_k + s_1 t_{k+1} + \ldots + s_{n-1} t_{k-1}$.

Решение

Решение:

- Представим s и t как многочлены с коэффициентами 0 и 1.
- Хочется сделать так, чтобы коэффициент у произведения многочленов был равен количеству совпадающих единиц.
- Пусть мы сдвинули t на k позиций вправо. Тогда ответ равен $s_0 t_k + s_1 t_{k+1} + \ldots + s_{n-1} t_{k-1}$.
- Перевернём многочлен для строки t — всё, кроме нулевого элемента.
- Пример: $s = 10110010$, $t = 01010111$.
- $S(x) = 1x^0 + 0x^1 + 1x^2 + 1x^3 + 0x^4 + 0x^5 + 1x^6 + 0x^7$.
- $T(x) = 0x^0 + 1x^7 + 0x^6 + 1x^5 + 0x^4 + 1x^3 + 1x^2 + 1x^1$.
- Теперь ответ для сдвига $k = 7$ — это $S_0 T_7 + S_1 T_6 + \ldots + S_7 T_0$, то есть коэффициент в произведении при x^7.

Решение

Решение:

- Пусть мы сдвинули t на k позиций вправо. Тогда ответ равен $s_0 t_k + s_1 t_{k+1} + \ldots + s_{n-1} t_{k-1}$.
- Перевернём многочлен для строки t — всё, кроме нулевого элемента.
- Пример: $s = 10110010$, $t = 01010111$.
- $S(x) = 1x^0 + 0x^1 + 1x^2 + 1x^3 + 0x^4 + 0x^5 + 1x^6 + 0x^7$.
- $T(x) = 0x^0 + 1x^7 + 0x^6 + 1x^5 + 0x^4 + 1x^3 + 1x^2 + 1x^1$.
- Теперь ответ для сдвига $k = 7$ — это $S_0 T_7 + S_1 T_6 + \ldots + S_7 T_0$, то есть коэффициент в произведении при x^7.
- А что с остальными? Например, при $k = 5$ ответ — это $S_0 T_5 + S_1 T_4 + S_2 T_3 + S_3 T_2 + S_4 T_1 + S_5 T_0 + S_6 T_7 + S_7 T_6$.

Решение

Решение:

- Пусть мы сдвинули t на k позиций вправо. Тогда ответ равен $s_0 t_k + s_1 t_{k+1} + \ldots + s_{n-1} t_{k-1}$.
- Перевернём многочлен для строки t — всё, кроме нулевого элемента.
- Пример: $s = 10110010$, $t = 01010111$.
- $S(x) = 1x^0 + 0x^1 + 1x^2 + 1x^3 + 0x^4 + 0x^5 + 1x^6 + 0x^7$.
- $T(x) = 0x^0 + 1x^7 + 0x^6 + 1x^5 + 0x^4 + 1x^3 + 1x^2 + 1x^1$.
- Теперь ответ для сдвига $k = 7$ — это $S_0 T_7 + S_1 T_6 + \ldots + S_7 T_0$, то есть коэффициент в произведении при x^7.
- А что с остальными? Например, при $k = 5$ ответ — это $S_0 T_5 + S_1 T_4 + S_2 T_3 + S_3 T_2 + S_4 T_1 + S_5 T_0 + S_6 T_7 + S_7 T_6$.
- Если делать свёртку ровно для n элементов, индексы получится взять по модулю автоматически!

Код

```
1       string s, t;
2       cin >> s >> t;
3       limit = int (s.size ());
4       logLimit = int (log2 (limit) + 0.5);
5       calcRev ();
6       calcZ ();
```

Ввод:

```
10110010
01010111
```

Вывод:

```
0: 2
1: 3
2: 2
3: 3
4: 2
5: 4
6: 1
7: 3
best = 4
pos = 5
```

Код

```
1       reverse (t.begin () + 1, t.end ());
2       vector <Num> a (limit), b (limit);
3       for (int i = 0; i < limit; i++) {
4           a[i] = Num (s[i] - '0');
5           b[i] = Num (t[i] - '0');
6       }
7
8       auto fa = fft (a);
9       auto fb = fft (b);
10      auto fc = vector <Num> (limit);
11      for (int i = 0; i < limit; i++)
12          fc[i] = fa[i] * fb[i];
13      auto c = fft (fc, true);
14      vector <int> res (limit);
15      for (int i = 0; i < limit; i++)
16          res[i] = int (c[i].real () + 0.5);
17
18      for (int i = 0; i < limit; i++)
19          cout << i << ": " << res[i] << endl;
20      auto pos = max_element (res.begin (), res.end ());
21      cout << "best = " << *pos << endl;
22      cout << "pos = " << pos - res.begin () << endl;
```

Ввод:

```
10110010
01010111
```

Вывод:

```
0: 2
1: 3
2: 2
3: 3
4: 2
5: 4
6: 1
7: 3
best = 4
pos = 5
```

Вопросы?

Вопросы?