Быстрое преобразование Фурье

$A, B \in \mathbb{R}[x]$ $\quad \mathbb{C}[x]$

$C = A \cdot B$

```
for i=0..degA-1
    for j=0..degB-1
        C[i+j]+=A[i]*B[j]
```

$$A \qquad\qquad\qquad\qquad B$$

1). $\downarrow$ $O(n^2)$ ✓ $O(n\log n)$ $\qquad \downarrow O(n^2)$

$$A(x_1), A(x_2), \dots A(x_n) \qquad B(x_1), \dots, B(x_n)$$

2). $C(x_i) = A(x_i)\cdot B(x_i)$ $\quad O(n)$

3). $\Big\downarrow$ интерпол. $\quad n \geq \deg C + 1 = \underbrace{\deg A + \deg B + 1}_{n :=}$

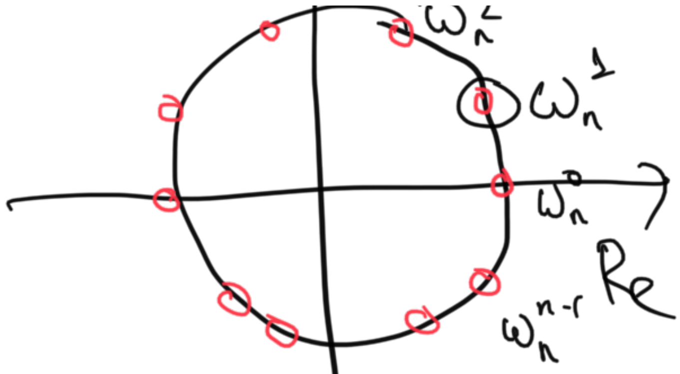
C $\quad O(n^2)$ ✓ $O(n\log n)$

Дискретн. преобр. Фурье (ДПФ, DFT)

Быстрое п. Ф. (FFT, БПФ)

Далее $n = 2^k$.

$$x_j = e^{\frac{2\pi i \cdot j}{n}}$$
$$j = 0..n-1$$



$$e^{\alpha i} = \cos\alpha + i\sin\alpha$$

$$\forall x_j \quad x_j^n = 1$$

$$\boxed{\begin{aligned} \omega_n &= e^{\frac{2\pi i}{n}} \\ x_j &= \omega_n^j \end{aligned}}$$

$$\omega_n^a = \omega_n^{a \% n}$$

```
def FFT(a):
    n = len(a)   // deg A = n-1
    if n == 1:
        return a
```

$$\boxed{(\omega_n^j)^2 = \left(e^{\frac{2\pi i \cdot j}{n}}\right)^2 = e^{\frac{2\pi i j}{n/2}} = \omega_{n/2}^j}$$
$$j < n/2$$

$$A(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} =$$
$$= (a_0 + a_2 x^2 + a_4 x^4 + \ldots) +$$
$$+ x(a_1 + a_3 x^2 + a_5 x^4 + \ldots) =$$
$$= \boxed{P(\underline{x^2}) + x \cdot Q(\underline{x^2})}$$

$$\mathrm{len}(P) = n/2 = \mathrm{len}(Q)$$

$$A(\omega_n^0), A(\omega_n^1), \ldots, A(\omega_n^{n-1})$$
$$P(\omega_{n/2}^0), P(\omega_{n/2}^1), \ldots, P(\omega_{n/2}^{n/2-1})$$
$$Q(\ldots)$$

$a \to p, q$ // чётн. и нечёт. коэфф.

$p = FFT(p)$

$q = FFT(q)$

$n/2$

$2k = n$

$k = n/2$

for $j = 0..n-1$:

$a[j] = \underbrace{p[j \% (n/2)]}_{P(\omega_{n/2}^{j}) = P(\omega_n^{2j})} + q[j \% (n/2)] \cdot$

$\cdot e^{\frac{2\pi i}{n} \cdot j}$

$\omega_n^j \quad \omega_n^{j+n/2} = \omega_n^j \cdot \underbrace{\omega_n^{n/2}}_{=-1}$

return a

$T(n) = O(n) + 2T(n/2) \Rightarrow T(n) = O(n \log n)$

Обратное преобразование

$Wa = v$

$$\underbrace{\begin{pmatrix} x_0^0 & x_0^1 & \dots & x_0^{n-1} \\ x_1^0 & x_1^1 & \dots & x_1^{n-1} \\ \vdots & & & \\ x_{n-1}^0 & \dots & & x_{n-1}^{n-1} \end{pmatrix}}_{W} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = v$$

$\boxed{a = W^{-1} v}$ n

$$W = \begin{pmatrix} \omega_n^{-0} & \omega_n^{-0\cdot 1} & \omega_n^{-0\cdot 2} & \dots \\ \omega_n^{-1\cdot 0} & \omega_n^{-1\cdot 1} & \omega_n^{-1\cdot 2} & \dots \end{pmatrix}$$

$$W^{-1} = \begin{pmatrix} \vdots & & & & \\ \omega_n^{-(n-1)0} & \omega_n^{-(n-1)\cdot 1} & \omega_n^{-(n-1)\cdot 2} & \dots & \omega_n^{-(n-1)(n-1)} \end{pmatrix}$$

$$W_{ij} = \omega_n^{ij} \qquad i,j \in \{0,\dots,n-1\}$$

$$W^{-1}_{ij} = \frac{\omega_n^{-ij}}{n}$$

$$\omega_n^{-k} = \omega_n^{n-k} = \overline{\omega_n^{k}}$$

$$F = C = W \times W'$$

$$C_{ij} = \sum_{k=0}^{n-1} W_{ik} \cdot W'_{kj} = \sum_{k=0}^{n-1} \omega_n^{ik} \cdot \frac{\omega_n^{-kj}}{n} =$$

$$= \frac{1}{n} \begin{cases} \sum_{k=0}^{n-1} \omega_n^0 = n & i=j \\ \sum_{k=0}^{n-1} \left(\omega_n^{i-j}\right)^k = \sum_{k=0}^{n-1} q^k = \frac{q^n-1}{q-1} = 0 \end{cases}$$

$$q = \omega_n^{i-j} \neq 1 \qquad i \neq_n j$$

$$\boxed{\begin{array}{l} C_{ii} = 1 \quad \forall i \\ C_{ij} = 0 \quad \forall i \neq j \end{array}}$$

$$\left(\omega_n^{(i-j)}\right)^n = \left(\omega_n^n\right)^{i-j} = 1^{i-j}$$

$$A(\omega_n^0), A(\omega_n^1), \dots \qquad A(\omega_n^{n-1})$$

$$A(\omega_n^0), \underbrace{A(\omega_n^{n-1}), A(\omega_n^{n-2}), \dots A(\omega_n^1)}$$

def inv_FFT(fa): // $fa = [A(\omega_n^0), \dots]$

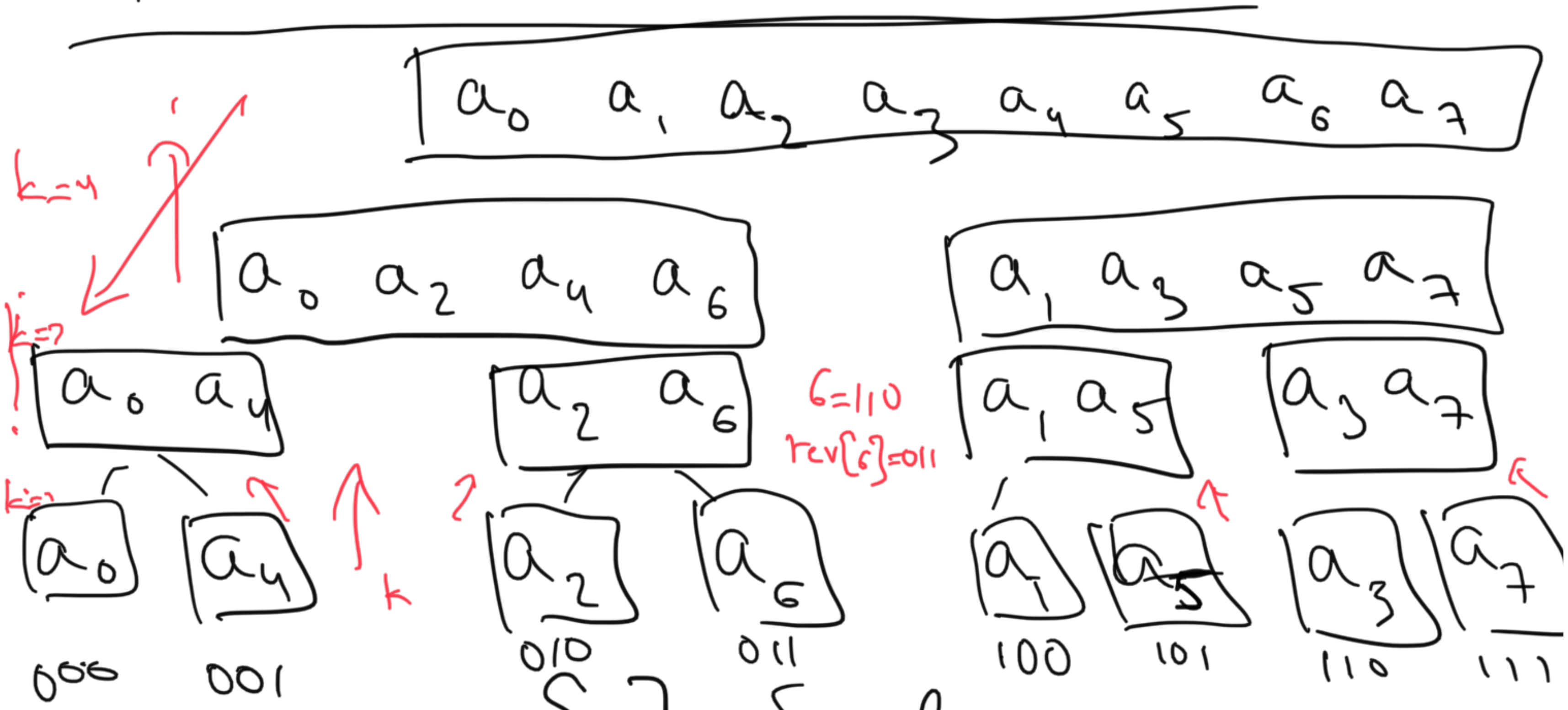
```
a = FFT(fa)                  ← O(n log n)
    reverse(a+1, a+n)        [1,n), нулев. не трог!
    for i = 0..n-1
        a[i] /= n            O(n)
    return a
```

$FFT(u) = W \cdot u$

$inv\_FFT(u) = W^{-1} \cdot u$

Wu

```
Mult(a, b)
    fa = FFT(a)
    fb = FFT(b)
    for i = 0..n-1
        fc[i] = fa[i] * fb[i]
    return inv_fft(fc)
```



rev[x] – битовое предст. в обр. порядке

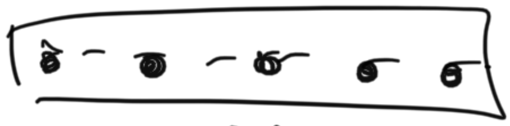
$n = 2^k$

```
void init(){
    for j = 0..n-1:
        rev[j] = rev[j/2]/2 + ((j%2) << (k-1))
```

| root[j] = $e^{\frac{2\pi i \cdot j}{n}}$

}

2n sin/cos



int rev[n];

$\underbrace{\text{complex<double>}}_{\text{num}}$ root[n];

for (k = 1; k < N; k *= 2)

  num tmp = $e^{\frac{2\pi i}{2k}}$;

  root[k] = num(1, 0);

k=1

  for (i = 1; i < k; ++i)

    root[k+i] = i%2==1 ? tmp * root[(k+i)/2] : root[(k+i)/2];

root $\underbrace{\omega_1^0}_{\omega_1}$ $\underbrace{\omega_2^0}_{\omega_2}$ $\underbrace{\omega_4^0\ \omega_4^1}_{\omega_4}$ $\underbrace{\omega_8^0\ \omega_8^1\ \omega_8^2\ \omega_8^3}_{\omega_8}$ . . .

[k, 2k) хранята корни ω_{2k}^{j} j=0..k-1

$\omega_{2k}^{2i} = \omega_k^i$

void FFT(a, fa) {

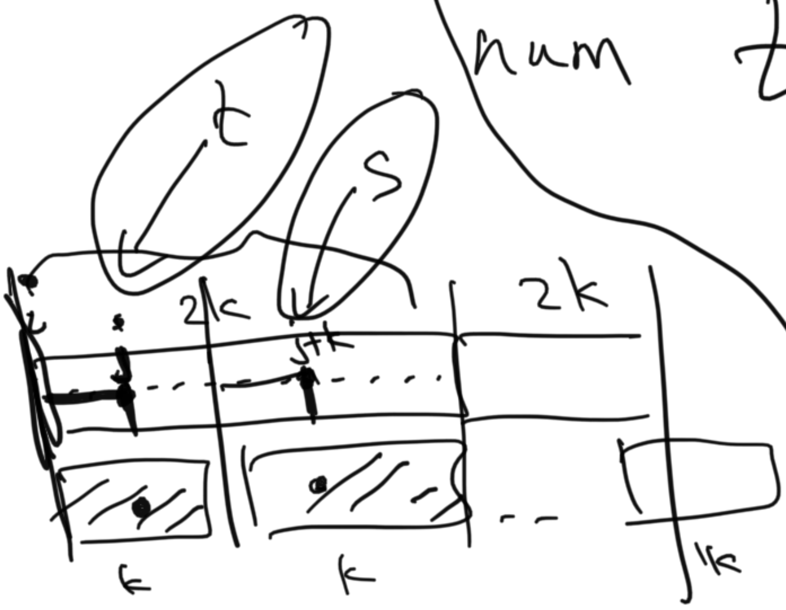
  for i=0..n-1

i=0..n-1

rev[i]=0..n-1

```
fa[rev[i]] = a[i]          init() д.б. заранее!
for (k = 1; k < N; k *= 2)          // блоки дл. k уже обр.
  for (i = 0; i < N; i += 2k)
    for (j = 0; j < k; ++j) {
      num tmp = root[k+j];  // ω_{2k}^j
```

$$\text{num } t = \underbrace{fa[i+j]}_{P[j\%(n/2)]} + tmp * \underbrace{fa[i+j+k]}_{q[j\%(n/2)]} \quad // \ fa[i+j]$$




```
      num s = fa[i+j] - tmp * fa[i+j+k]
      fa[i+j] = t
      fa[i+j+k] = s
    }
}
```

$3 \to 2$

$$A \xrightarrow{FFT} FA \qquad B \xrightarrow{FFT} FB$$

$$\mathrm{FC} \xrightarrow{(i)\mathrm{FFT}} C$$

$$A, B \in \mathbb{R}[x]$$

$$D = A + i \cdot B \qquad d_j = \underbrace{a_j}_{\in \mathbb{R}} + i \cdot \underbrace{b_j}_{\in \mathbb{R}}$$

$$FD = FFT(D)$$

$$D(\omega_n^0), \ldots, D(\omega_n^{n-1})$$

$$\underbrace{D(x) + \bar{D}(x)}_{= (D+\bar{D})(x)} = A(x) + A(x) = 2A(x)$$

$$A(x) = \frac{D(x) + \boxed{\bar{D}(x)}}{2}$$

$$B(x) = \frac{D(x) - \boxed{\bar{D}(x)}}{+2i}$$

$$\frac{z + \bar{z}}{2} = \operatorname{Re} z$$

$$\sum d_j x^j + \sum \bar{d}_j x^j$$

$$\frac{\sum a_j x + \sum a_j \bar{x}}{2}$$

$$\boxed{\text{Утв. } \overline{D(x)} = \overline{D}(\bar{x})}$$

$$\overline{D(\bar{x})} = \overline{D}(x) \qquad x = \omega_n^{j}, \quad \bar{x} = \omega_n^{n-j}$$

$$\overline{FD[n-j]}$$

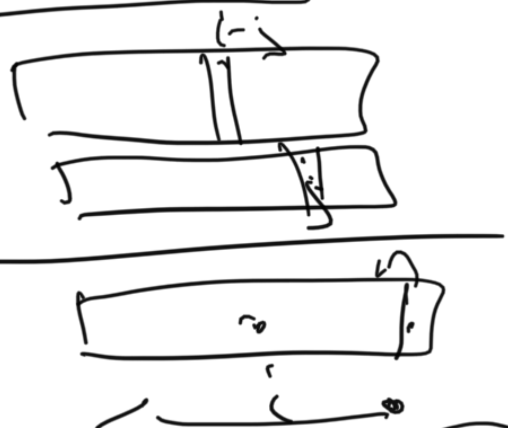
$$FA[j] = \frac{FD[j] + \overline{FD[n-j]}}{2}$$

$$FB[j] = \dots$$

Умножение чисел

$$\boxed{c[i] = \sum_{j=0}^{i} a[j] * b[i-j]}$$

$\oplus$ переносы



- основание 10
- основание 10^k

$$O\left(\frac{n}{k} \log \frac{n}{k}\right) \qquad n = 10^6 \qquad n^{\log_2 3}$$

$|c[i]| \le \underline{n \cdot 10^{2k}} = 10^6 \cdot 10^{2k}$

точность $c[i]$ не меньше 0.5

double $\to$ $k \le 4$
~15

long double $\to$ $k \le 6$
~18