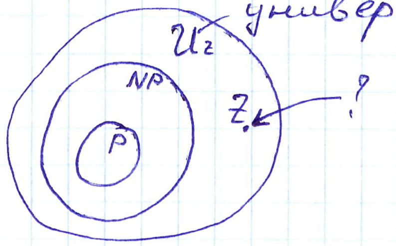
4. Класс NP

$Z \in NP \Leftrightarrow Z_v \in P$

$\exists A_v : f^{\wedge}_{A_v}(n) = O(n^k)$

5. Состояние до 1971 г.

универсум задач Z

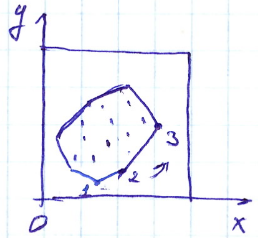


$P \subset NP$

P - полиномиально решаемые задачи

NP - полиномиально проверяемые задачи

6. Сведение

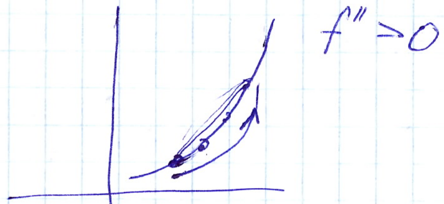


$\{(x, y)\}$

$a_1, a_2, \ldots, a_n$

$a_i \to (a_i ; a_i^2)$

любая возрастающ. ф-ция



Z_1, Z_2 $\qquad Z_1 \underset{P}{\Rightarrow} Z_2$

Z_1 полиномиально сводима к задаче Z_2

$D^{(1)} \in Z_1 \to D^{(2)} \in Z_2$

$D^{(1)} \xrightarrow[A_s]{} D^{(2)}$

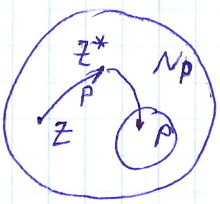
$Z_s(Z_1, Z_2) \in P \quad \exists A_s : f^{\wedge}(n) = O(n^k)$

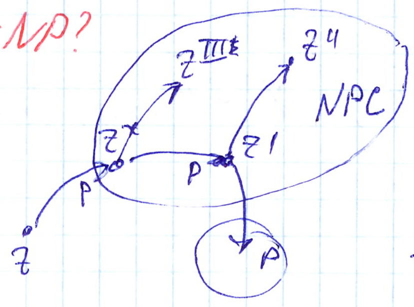
7. Теорема С. Кука 1971

$\exists Z^* \in NP : \forall Z \in NP$

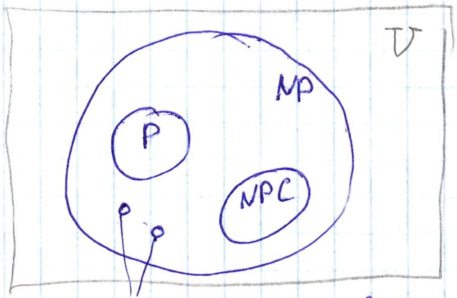
$Z \underset{P}{\Rightarrow} Z^*$

8. Проблема P=NP?





$f_{thlim} > n^k \Rightarrow P \neq NP$



Осталось 2 задачи

17.11.17

Класс NPC

1. Теорема Кука (1976)

$\exists z^* \in NP : \forall z \in NP \quad z \underset{P}{\Rightarrow} z^*$

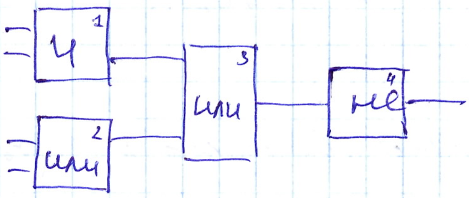
2. Задача о выполнимости схемы (z^*)

а) Функциональная схема

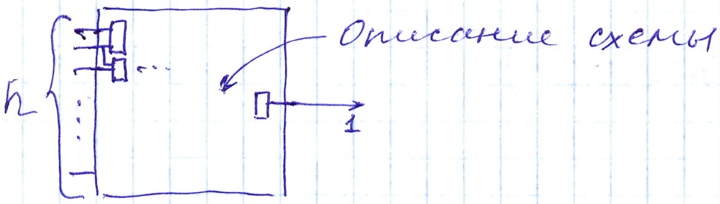
- Базис (и, или, не)

└→ функц. эл-ты

в) Описание схемы - язык



с) Фиксация вида схемы



Мощность схемы $O(n^k)$

$z^* : D \in D_n \quad D = \{n, \underset{O(n^k)}{\text{описание}} \text{ схемы}\}$

$A_{z^*} \quad D \in D_n \longrightarrow X_n$

$X_n = (x_1, x_2, \dots, x_n) \quad X_n$: Выход схемы = 1

3. Класс NPC

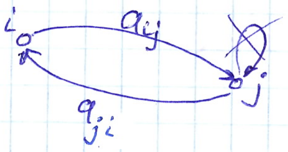
1971 → z* NPC – NP-complete

1979 Гэри, Джонсон „Вычислительные машины и трудно решаемые задачи"

Сегодня

4. NP-трудные

ATSP – асимметричная задача коммивояжера



Тур – гамильтонов цикл (проход. по кажд верш 1 раз)

$ATSP = K_n$

$T^*: C(T^*) \to \min\limits_T$

$|T| = (n-1)!$

! Преобразование задач оптимизации (дискр, комбин) в задачи „разрешения"

$T^* \to \min \Rightarrow \exists T^\#: \underbrace{C(T^\#) < T_0}_{O(n^1)}$

NP-трудные задачи – задачи, которые, будучи переформулированы задачи разреш. нах-ся в NP.

5. ε-полиномиальные приближения

1971 → 1975 ⟶ 1996 (?) ⟶ ~

Генетический алгоритм — Муравьиный — Пчелы

$A: T' : |T' - T^*| < \varepsilon$

1998 ал-м Ароры

24.11.17.

Рекурсивные функции

1. Три ипостаси термина

- математика (теория рекурсии)
- рекурс. функции Черч, Клини (Теор. алг.)
- рекур. функции (программир)

2. Рекурсивные функции в математике

$f: Z^+ \xrightarrow[f]{} Z^+ \qquad Z^+ = N \cup "0"$

Рекурсия - определение функции через предыдущие значения

$f(0) = f_0, \quad f(1) = f_1, \ldots,$

$f(k) = g(f(k-1), \ldots, \ldots)$

Запись

$$\begin{cases} f(0) = f_0 \\ f(1) = f_1 \\ \vdots \\ f(k) = g(f(k-1), \ldots; k, const) \end{cases} \Rightarrow \text{Рекуррентное соотношение}$$

Решить рекуррентное соотношение - получить аналитическую функцию $f(k) = \underbrace{h(k)}_{\text{аналитический вид}}$

3. Примеры

а) $\begin{cases} f(1) = 1 \\ f(k) = f(k-1) + 1, \quad k \geq 2 \end{cases} \qquad f(k) = k$

б) $\begin{cases} f(1) = 1 \\ f(k) = f(k-1) + 2k - 1 \end{cases} \qquad f(1) = 1 \quad f(2) = 4 \quad f(3) = 9 \qquad f(k) = k^2$

c) $\begin{cases} f(0) = 1 \\ f(k) = 2f(k-1) \end{cases}$ $f(k) = 2^k$

d) $\begin{cases} f(0) = 1 \\ f(k) = \sum_{i=0}^{k-1} f(i) + 1 \end{cases}$ $f(k) = 2^k$

e) $\begin{cases} f(1) = 1 \\ f(2) = 1 \\ f(k) = f(k-1) + f(k-2) \end{cases}$ 1 1 2 3 5 8 13 21

$f(k) \approx \varphi^k \approx 1,618^k$

f) $\begin{cases} f(0) = 1 \\ f(k) = k \cdot f(k-1) \end{cases}$ $f(k) = k!$

$$\Gamma(s) = \int_0^{\infty} x^{s-1} e^{-x} dx$$

$$\Gamma(s+1) = s\Gamma(s)$$

$$n! = \Gamma(n+1)$$

$$0! = \Gamma(1) = 1$$

$$3,86! = \Gamma(4,86)$$

g)! $\beta_1(n)$

$n_{(10)}$	1	2	3	4	5	6	7	8	9
$n_{(2)}$	0001	0010	0011	0100	0101	0110	0111	1000	1001
$\beta_1(n)$	1	1	2	1	2	2	3	1	2

подсчет число единиц в двоичном представлении числа n

$$\beta_1(2^k) = 1$$

$$\beta_1(2^k - 1) = k$$

$$\beta_1(n-1) \le \log_2(n+1)$$

$$\beta_1(n) \le \lceil \log_2 n \rceil$$

$$\begin{cases} \beta_1(0) = 0 \\ \beta_1(1) = 1 \\ \beta_1(2k) = \beta_1(k) \\ \beta_1(2k+1) = \beta_1(k) + 1 \end{cases}$$

4. Программирование

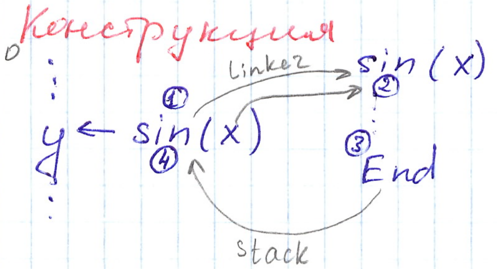
Факториал

$$\begin{cases} f(0)=1 \\ f(n)=n\cdot f(n-1) \end{cases} \Rightarrow$$

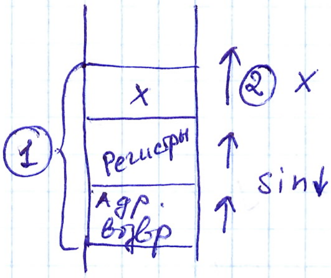
```
Function F(n)
   if n=0
     then
        F ← 1
   else
        F ← n·F(n-1)
End.
```

Механизм вызова функции

1. Конструкция



2. Программный стек.



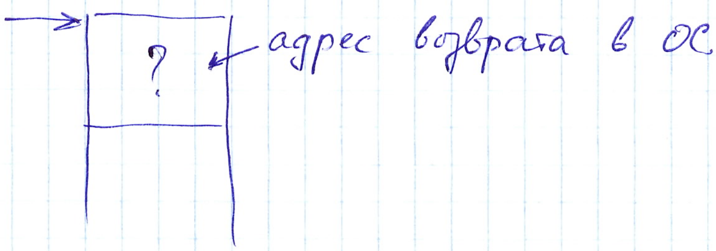
sin↓ ③ $\quad$ $\text{sin}\overset{y}{\uparrow}$ ④

3. Трудоемкость Call/Return

$$f_{C/R}(1) = 2(\underset{\substack{\text{число}\\\text{парам}}}{m} + \underset{\substack{\text{число}\\\text{регистров}}}{2} + \underset{\substack{f\\\text{знач}\\\text{ф-ции}}}{1} + \underset{\substack{\text{адр}\\\text{возвр}}}{1}) + \underset{\text{goto}}{2}$$

(обычно r=1, т.к. регистры пишутся одной командой)

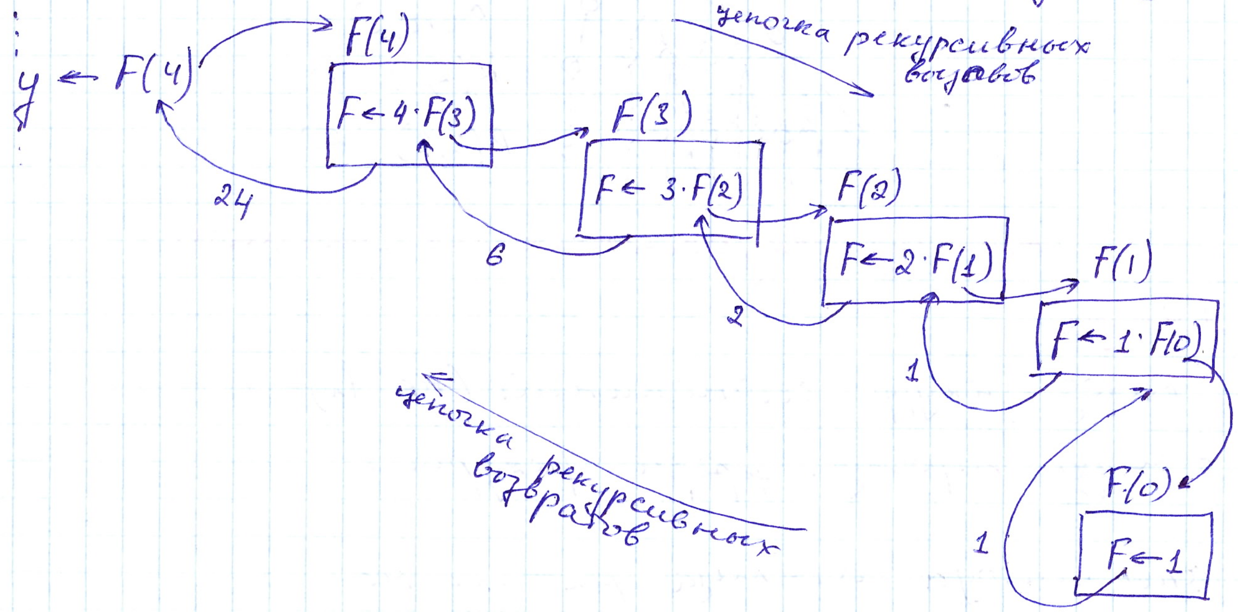
4.

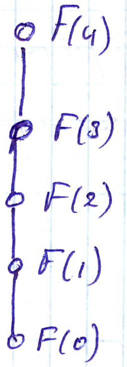


Порожденное дерево рекурсии.

1.12.17

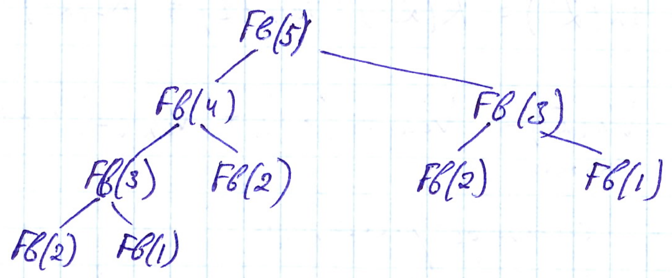
Рассмотрим ф-цию факториал (см. пред. лек.)





Унарное дерево

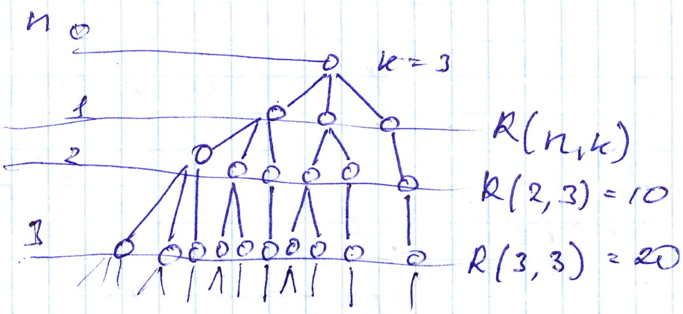
Fb(k)
  If ...
  else
    Fb ← Fb(k-1) + Fb(k-2)



$R_A(D)$ (или $R(D)$) – Общее число вершин

$R_V(D)$ – число внутренних вершин (те, у которых ещё идут рекурсивные вызовы)

$R_L(D)$ – число листьев



Анализ рекурсивных алгоритмов

1. Методы

- метод рекурр. соотношений (на f_A)
- метод порожд. дерева рекурсии

Этапы метода анализа рек алг-ма

1. $f_L(1)$ - трудоёмкость в листе
2. $f_V(1)$ - трудоёмкость во внутренней вершине
3. Исследование дерева (получение R, R_L, R_V, H_{max})

   R, R_L, R_V, H_{max} - высота дерева
4. Общая формула

   2'. $f_{C/R}(1)$ - вызов / возврат на 1

$$f_A(D) = R(D) \cdot f_{C/R}(1) + R_L(D) \cdot f_L(1) + R_V(D) \cdot f_V(1)$$

$$V_A(D) = V_{собств}(D) + V_{C/R}(1) \cdot H_{max}(D)$$

память

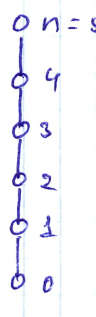
Для факториала

$f_L(1) = 2$ (сравнение + запись)

$f_V(1) = 4$ (сравн, выч, умн, присвоение)

$f_{C/R}(1) = 2\left(\underset{m}{1} + \underset{r}{1} + \underset{\text{адр. возвр}}{1} + \underset{F}{1}\right) + 2 = 10$

регистры пишутся одной командой



$R(n) = n + 1$

$R_V(n) = n$

$R_L(n) = 1$

$f_A(n) = (n+1) \cdot 10 + 1 \cdot 2 + n \cdot 4 = 14n + 12$

„Методы" разработки

- метод декомпозиции
- метод динамического программирования

Метод декомпозиции

8.12.17.

1. Идея

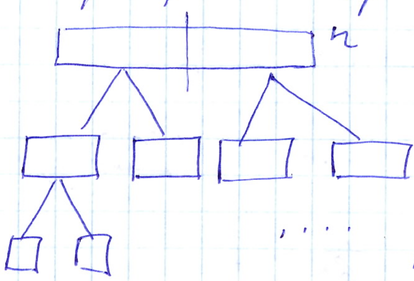
Z - разм n

- деление на части до n_0
- объединение решений

! ~~Целевой~~ Только для Z, целевой функционал которых аддитивен

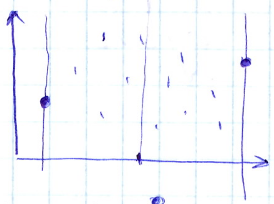
2. Примеры

а) Сортировка (ф. Нейман)



$n_0 = 1$

в) Контур



$n_0 = 3$ △

A

A_1	A_2
A_3	A_4

B

B_1	B_2
B_3	B_4

=

C_1	

$C_1 = A_1 \cdot B_1 + A_2 B_3$ $n_0 = 2$

3. Трудоёмкость

$$\left.\begin{array}{l} - d(n) - \text{деление} \\ - u(n) - \text{объединение} \end{array}\right\} d(n) + u(n) = g(n)$$

$d(n) \to a$ подзадач разм n/b $\left(\lfloor \frac{n}{b} \rfloor, \lceil \frac{n}{b} \rceil\right)$

$$\Rightarrow \begin{cases} f_A(n) = a\cdot f_A(n/b) + g(n) \\ f_A(n_0) = c_0 \end{cases}$$

4. Теорема (Дж Бентли, Д. Хаакен, Дж Сакс 1980)

$f_A(n) = a\, f_A(n/b) + g(n)$ ← трудоемкость в листе

$\Downarrow$ дерево рекурсии

$n^{\log_b a}$

ε – полиномиальный зазор

I. $n^{\log_b a - \varepsilon} \succ g(n)$, то $f_A(n) = \Theta(n^{\log_b a})$

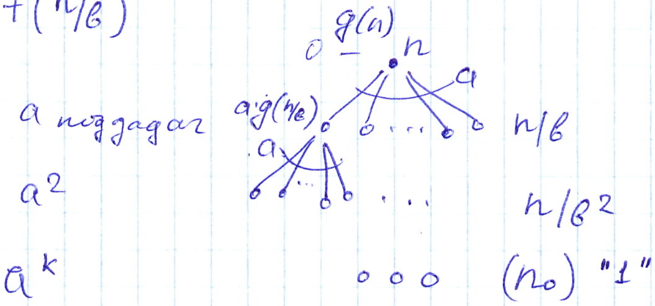
II. $g(n) \succ n^{\log_b a + \varepsilon}$ + усл. регул $\exists\, c>0: \; a\, g(n/b) = c\, g(n)$

то $f_A(n) = \Theta(g(n))$

III. $n^{\log_b a} = \Theta(g(n))$

$f_A(n) = \Theta(n^{\log_b a} \cdot \ln n)$

$a f(n/b)$



к уровней

$\frac{n}{b^k}$ – число листьев на уровне

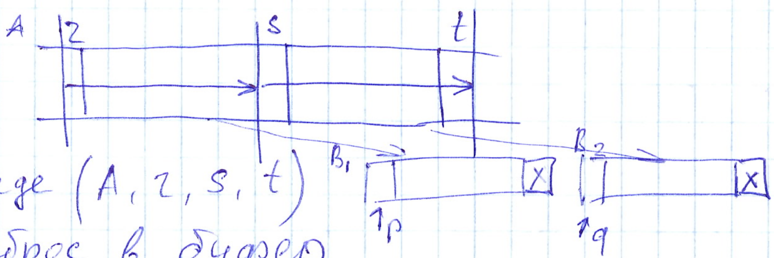
$\frac{n}{b^k} = 1$

$n = b^k$

$k = \log_b n$

$a^k = a^{\log_b n} = n^{\log_b a}$

5. Сортировка слиянием



$V(n) = \Theta(1)$

$\Rightarrow f(n) = \Theta(n)$

Merge (A, r, s, t)

- сброс в буфер

$k \leftarrow r$

for ...

If $B1[p] < B2[q]$

then $A[k] \leftarrow B1[p]$

$p \leftarrow p+1$

else ...

$18n + 23$

$f_1(n) = 18 n \log_2 n + 91 n + \ldots$

Умножение длинных целых

15.12.17.

1. Представление

A

1		n
0/1	0/1	

B

$A \times B = C \qquad M(D) = n$

2. „Столбик"

$$\begin{array}{r} \times\ 101 \\ 111 \\ \hline 101 \\ 101 \\ 101 \\ \hline 100011 \end{array}$$

Алгоритмы

I. Прямая реализация – $n^2 + V(n) = \Theta(n^2)$

II. Посл. сложение

$C \leftarrow A * B[n]$

$C \leftarrow C + A * 2^k * B[n-1]$

модиф.

$$\begin{array}{r} 111 \\ 101 \\ \hline 111 \\ 111 \\ \hline 100011 \end{array} \qquad \leftarrow B_1(B) \sim \frac{n}{2}$$

3. Алгоритм Карацубы

A

AH	AL

n

$A = AH \cdot 2^{n/2} + AL$

2^n реализ. как сдвиг на n позиций

B

BH	BL

n

$B = BH \cdot 2^{n/2} + BL$

$C = AH \cdot BH \cdot 2^n + (AH * BL + AL * BH) \cdot 2^{n/2} + AL * BL$

$f_A(n) = 4 f_A(n/2) + \underbrace{(Cn + b)}_{g(n)}$

$n^{\log_2 4} = n^2$

$$(AH+AL)(BH+BL) = AH\cdot BH + AL*BH + AH*BL + AL\cdot BL$$

$$C = AH\cdot BH\cdot 2^n + [(AH+AL)\cdot(BH+BL) - AH\cdot BH - AL\cdot BL]\cdot 2^{n/2} +$$
$$+ AL\cdot BL$$

$$f_A(n) = 3 f_A(n/2) + c\cdot n$$

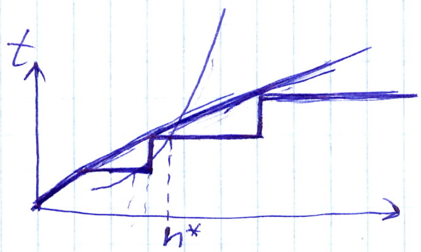
$$n^{\log_2 3} \approx n^{1,58xx}$$

Реализация

$$n \to \tilde{n} = 2^{\lceil \log_2 n \rceil}$$

437 → 512
624 → 1024

$$f_A(n) = O\left[\left(2^{\lceil \log_2 n \rceil}\right)^{\log_2 3}\right]$$



4. W. Straßen, A. Schönhage (1969)

a. Представление полиномов

$$g(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_0$$

I. Кортеж $A = (a_{n-1}, \dots, a_0)$

II. Кортеж значений $(g(x_0), g(x_1), \dots, g(x_{n-1}))$

III. Нули

в. Умножение полиномов

I. Кортеж коэфф.

$$\underbrace{(a_2x^2 + a_1x + a_0)}_{a(x)}\cdot\underbrace{(b_2x^2 + b_1x + b_0)}_{b(x)} =$$

$$= a_2b_2x^4 + (a_2b_1 + a_1b_2)x^3 + (a_2b_0 + a_1b_1 + a_0b_2)x^3 +$$

A, B $C \Rightarrow g(x)\cdot h(x)$

$C = A*B$

II. Кортеж значений

$f(x) = g(x)\cdot h(x)$

$f(x_0) = g(x_0)\cdot h(x_0)$

! $g(x)$ и $h(x)$ заданы в $2n$ точках

c. <u>Переходы представлений</u>

$I \to II$ Схема Горнера $g(x_i) \Rightarrow \Theta(n)$

$\forall x_i \; g(x_i) \to \Theta(n^2)$

$II \to I$

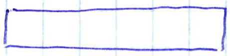
$$\begin{cases} a_{n-1} x_0^n + \ldots + a_0 = g(x_0) \\ \vdots \\ a_{n-1} x_{n-1}^n + \ldots + a_0 = g(x_{n-1}) \end{cases} \Rightarrow \Theta(n^2)$$

d. <u>Идея</u>

A 

$\Rightarrow g(x) = a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_0 \Rightarrow II$

$g(2) = A$

B 

$\Rightarrow h(x) = b_{n-1} x^{n-1} + \ldots + b_0 \Rightarrow II$

$\forall_i \; f(x_i) = g(x_i) \cdot h(x_i) \to I$

e. 1964(5) Кули и Тьюки.

DFT $(a_{n-1}, \ldots, a_0) \Rightarrow a_{n-1} x^{n-1} + \ldots = g(x)$

$\Downarrow \Theta(n^2)$

$(g(\omega_i) \ldots)$

FFT $(\ldots g(\omega_i) \ldots) \; \Theta(n \ln n)$

$C = FFT^{-1}_{2n}(\underbrace{FFT_{2n}(A)}_{\substack{g(\;) \\ 2n \ln 2n}} \underbrace{*}_{\Theta(2n)} FFT_{2n}(B))$

$f_A(n) = O(n \ln n \ln \ln n)$