Анализ алгоритмов

Ульянов Михаил Васильевич

2019

Оглавление

Глава 1

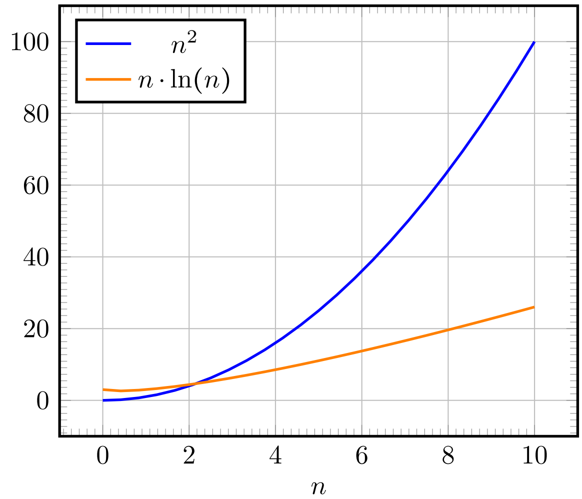
Исторический очерк

1. **1900**
   Д. Гильберт - 23 проблемы 1931 - К. Гедель доказал теорему о неполноте

2. **1936**
   А.Тьюринг, Э.Л.Пост - Теория алгоритмов (начало)

   - формализация понятия
   - общие свойства
   - обнаружение алгоритмически неразрешимых задач

3. **1960е**
   Теория сложности вычислений NPC
   $O(n^2)$ $O(n \cdot \ln(n))$



4. **Начало 1970х**
   Практический анализ алгоритмов Д.Э. Кнут

Глава 2

Схема выбора алгоритмического обеспечения

Нет:

A. Новый (метод разработки)

B. Комбинированные элементы $(A_1 + A_2 + A_3)$

$$Q(q_1, ..., q_m) = \sum \alpha_i q_i \to R^1 \text{ - комплексные оценки}$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases}$$

$$a + ib = (c + id) =^{det} (ac - bd) + i(bc + ad)$$

Глава 3

1936 год Э.Л. Пост "Финитные комбинаторные процессы - формулировка 1"

3.1 Терминология

Общая проблема = задача
Конкретная проблема = индивидуальная задача

3.2 Общая проблема

Общая → множество всех конкретных
Решение общей → решение каждой конкретной

3.3 Пространство символов

←							→

	Помечен, не помечен	

Конкретная проблема задается внешней силой путем пометки конечного числа символов.

3.4 Работник (процессор)

1. → (R)
2. ← (L)
3. ✓ Поставить метку, если пусто
4. ξ Стереть, если есть
5. ? $\xrightarrow{\text{да}} N^o$ строки
$\xrightarrow{\text{нет}} N^o$ строки
6. stop

3.5 Примеры

	✓	✓	✓	✓	...	✓	

1. ξ
2. →

3. $? \begin{array}{l} \overset{\text{да}}{\to} 1 \\ \overset{\text{нет}}{\to} 4 \end{array}$

4. stop

3.6 Гипотеза Поста

a) Программа применима к общей, если $\forall$ конкретной нет коллизий в операциях 3,4

b) программа заканчивается, если stop

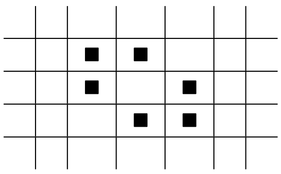
c) Если $\forall$ конкретной внеш сила распознает правильный ответ, то Ф1П есть 1-решение общей

d) Мы вправе рассматривать все более и более широкие формулы

пространство символов	
алфавита	логически сводимы к формуле 1
набор инструкций	

a + b = Финитный 1 процесс

3.7 1984 Муравей Лэнгтона



$$W \to (B, R)$$

$$B \to (W, L)$$

Глава 4

Терминология

4.1 Вход алгоритма

Конкретная проблема $\Rightarrow$ индивидуальная задача $\Rightarrow$ Вход $D = \{d_i | i = 1, m\}$

4.2 Длина входа

$n \to |D|$?

$$\mu_z(D) \to g(|D|)$$

$$\mu_{sort}(D) = |D| - 1$$

Умножение матриц $n \times n$

$$|D| = 2n^2 + 1$$

$$\mu_z(D) = n = \sqrt{\frac{|D| - 1}{2}}$$

4.3 Трудоемкость

$f_A(D)$ - число элементарных операций принятой модели вычислений заданных алгоритмом A на входе D

$$f_{A_1}(D) < F_{A_2}(D)$$

4.4 D_n

$$D_n = \{D | \mu_z(D) = n\}$$

$$\beta = 16 \quad |D_{10}| = 2^{160}$$

4.5 Переход к n

$$f_A^{\vee}(n) \overset{def}{=} \min_{D \in D_n} f_A(D) \text{ - лучший случай}$$

$$f_A^{\wedge}(n) \overset{def}{=} \max_{D \in D_m} f_A(D) \text{ - худший случай}$$

$$\overline{f}_A(n) = \sum_{D \in D_n} p(D) - f_A(D)$$

$$p(D) = \frac{1}{|D_n}$$

4.6 Память

$$V(D) = \underbrace{|D|}_{\text{Вход}} + \underbrace{|R|}_{\text{Результат}} + V_{\text{дополнительно памяти}}(D) + V_{\text{программы}}$$

$$V_A(D) = V_{\text{доп}}(D)|V^{\vee}(n), V^{\vee}(n), ...$$

Глава 5

Примеры

5.1 Умножение матриц

```
MultM(A,B,n;C)
    for i ← 1 to n
        for j ← 1 to n
            s ← 0
            for k ← 1 to n
                s ← s + A[i,k] * B[k,j]
            c[i,j] ← s
End
```

$$f_A^{\vee}(n) = f_A^{\wedge}(n) = f_A(n) = 1 + n(3 + 1 + n(3 + 2 + n(3 + 7) + 3)) = 10n^3 + 8n^2 + 4n + 1$$

5.2 Max

```
Max(A,n;M)
    M ← A[1]
    for j ← 2 to n
        if M < A[i]
            M ← A[i]
End
```

$$f_A^{\vee}(n) = 2 + 1 + (n - 1)(3 + 2) = 5n - 2$$

$$f_A^{\wedge}(n) = 2 + 1 + (n_1)(3 + 2 + 2) = 7n - 4$$

5.3 Степень

```
Pow(x,k;y)
    ...
    y → 1
    for j → 1 to k
        y → y * x
End
```

$$D \in D_2$$

$$M_z(D) = 2$$

$$f_A(k) = 2 + 5k$$

$$e^x = e^{\lfloor x \rfloor} \cdot e^{\{x\}}$$

$$e^\alpha = \sum^{\infty} \frac{\alpha^k}{k!}, 0 \le \alpha < 1, \frac{\alpha^k}{k!} > \sum_{k+1}^{\infty} (.)$$

$$\mathrm{Exp}(x, eps, y) \Rightarrow k^* = g(eps) \quad k^* : \frac{\alpha^{k^*}}{k^*!}$$

Глава 6

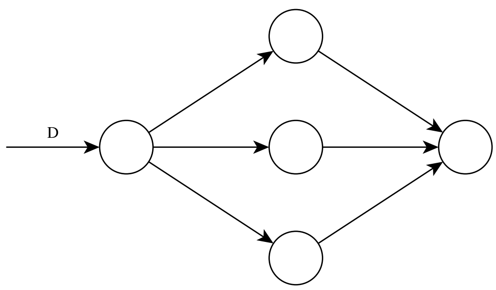
Классификация алгоритмов по типу трудоемкости

6.1 Класс N - количесвенно-зависимые алгоритмы

$$f_A(D) = f_A(\mu_Z(D)) = f_A(n)$$

$$f^{\vee} = f^{\wedge}!$$

!Матрично-векторные операции



Гипотеза П.Леви

$$R^n \ \overline{a} = (..., [-1, 1], ...)$$

6.2 Класс PR - параметрически-зависимые алгоритмы

$$f_A(D) = f_A(P + 1, ..., p + k) \neq f(n)$$

$$li(x0) = \int_2^x \frac{1}{\ln x} dx$$

6.3 Класс NPR

$$f_A(D) = f_A(n, pr)$$

6.4 Декомпозиция f в NPR

$$f_A(D) = f_n(n) \underset{*}{+} g_{PR}(D)$$

$$f^{\wedge}_A(n) = f_n(n) + g^{\wedge}_{PR}(n)$$

6.5 Асимптотическая иерархия функций

$$a < b$$

$$f(x) \Rightarrow f(\cdot)$$

$$f(x) \prec g(x)$$

$$\prec: \lim_{x \to \infty} \frac{f(x)}{g(x)} = 0$$

$$\ln x \prec \sqrt{x} \prec \underset{k > \frac{1}{2}}{x^k} \prec e^{\lambda x} \prec x^x$$

$$f(x) \asymp g(x) \Rightarrow \lim_{x \to \infty \frac{f}{g} \neq 0}$$

6.6 Подклассы в NPR

$$f_A^{\wedge}(n) = f_n(n) + g^{\wedge}(n)$$

6.6.1 Подклассы NPR(Low)

$$g^{\wedge}(n) \prec f_n(n)$$

6.6.2 NPRE (Eq)

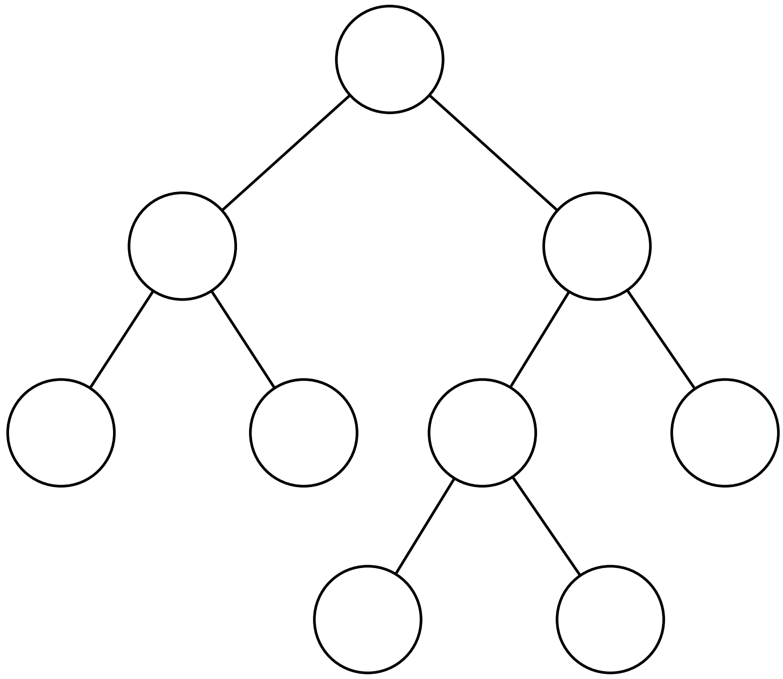
$$g^{\wedge}(n) \asymp f_n(n)$$

6.6.3 NPRH

$$f_n(n) \prec g^{\wedge}(n)$$

$$!g^{\wedge} \to \overline{g}$$

6.6.4 TSP



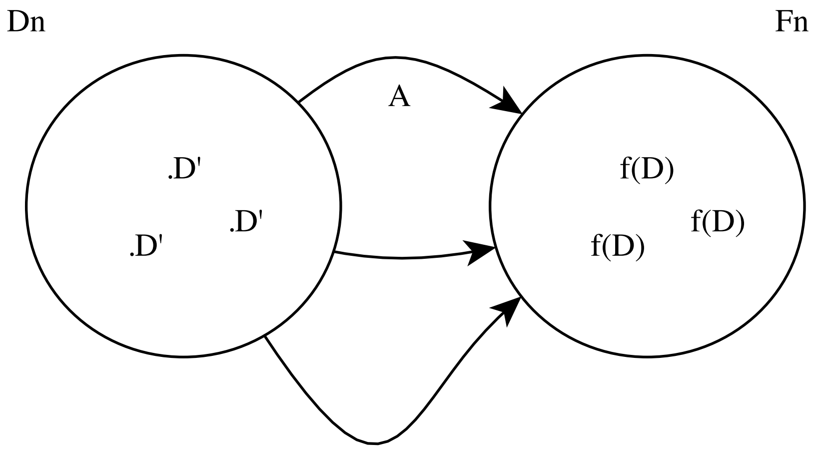
Глава 7

Метод классов эквивалентности

7.1 Математические сведения

1. Отношение R на A $R \subset A \times A$
2. Если R - отношения эквиваленстности:
   (a) $(a,a) \in R$ - рефлекс
   (b) $(a,a') \in R \Rightarrow (a',a) \in R$ - симметричность
   (c) $(a,a') \in R, (a',a'') \in R \Rightarrow (a,a'') \in R$ - транзитивность
3. Принцип Дирихле P.G.L. Dirichle

7.2 Идея метода



$$D_n \underset{A}{\to} F_n$$

$$f^{\vee}(n)$$

$$\exists D : f(D) = f^{\vee} + 1$$

$$R \text{ на } D_n (D,D') \in R \Leftrightarrow f_A(D) = f_A(D')$$

$$f^{\vee}, f^{\wedge} P(f(D) = f_1) = \frac{|D_{f_1}|}{|D_n|}$$

$$\overline{f_A}(n) = \sum_k f_k \cdot P(f = f_K)$$

7.2.1 Реализация

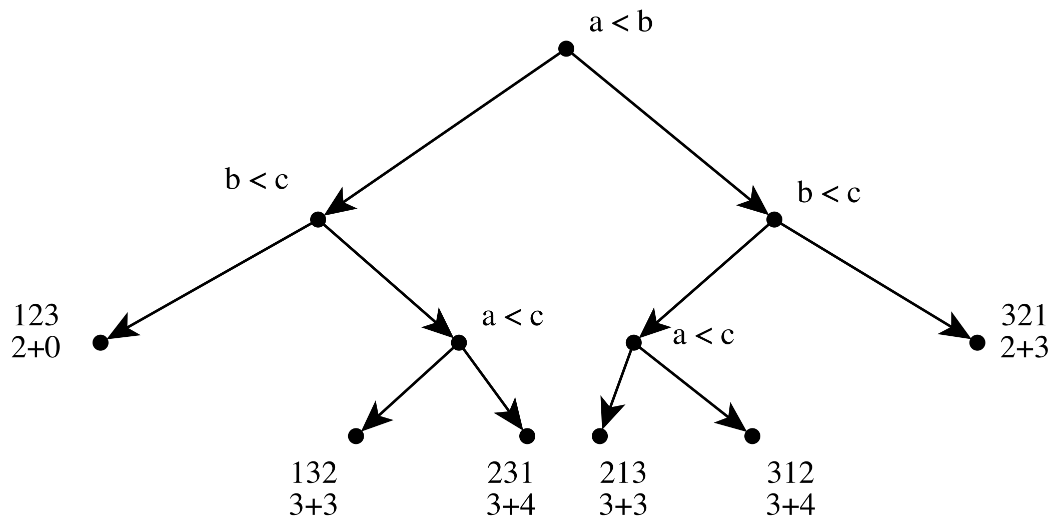
1. Равномощное разбиение
2. Объединение в классы

7.3 Сортировка 3х чисел по методу

```
Sort3(a,b,c) ← D_3
    if a > b
        then a ←→ b
    if b > c
        then b ←→ c
    if a > b
        then a ←→ b
End
```

a, b, c			f	$f^{\vee}$
1 - min		D_{123}	3	
2 - midl		D_{132}	6	
3 - max	$\Rightarrow$	D_{213}	6	$\overline{f} = 7\frac{1}{2}$
		D_{231}	9	
		D_{312}	9	
		D_{321}	12	

7.3.1 Модификация



	$f^{\vee}$	
D_{123}	2	
D_{132}	6	
D_{213}	6	$\overline{f} = 5\frac{1}{2}$
D_{231}	7	
D_{312}	7	
D_{321}	5	

7.4 Анализ max

```
M ← A[1]
for j ← 2 to n
    if M < A[j]
        then M ← A[j]
```

$$D_4 \to n = 4$$

1 - min	2 1 3 4 qqqq
2 - min2	2 1 4 3
3 - max2	2 3 1 4
4 - max	2 3 4 1

$$g_n(x) = x^{\overline{n}}$$

$$g_4(x) = x(x+1)(x+2)(x+3)$$

$$g_4(1) = 24$$

$$g_4(x) = 1 \cdot x^4 + 6 \cdot x^3 + 11 \cdot x^2 + 6 \cdot x$$

$$S_n^k$$