

Анализ алгоритмов

Ульянов Михаил Васильевич

2019

Оглавление

1	Исторический очерк	2
2	Схема выбора алгоритмического обеспечения	3
3	1936 год Э.Л. Пост "Финитные комбинаторные процессы - формулировка 1"	4
3.1	Терминология	4
3.2	Общая проблема	4
3.3	Пространство символов	4
3.4	Работник (процессор)	4
3.5	Примеры	4
3.6	Гипотеза Поста	5
3.7	1984 Муравей Лэнгтона	5
4	Терминология	6
4.1	Вход алгоритма	6
4.2	Длина входа	6
4.3	Трудоемкость	6
4.4	D_n	6
4.5	Переход к n	6
4.6	Память	7
5	Примеры	8
5.1	Умножение матриц	8
5.2	Мах	8
5.3	Степень	8
6	Классификация алгоритмов по типу трудоемкости	10
6.1	Класс N - количественно-зависимые алгоритмы	10
6.2	Класс PR - параметрически-зависимые алгоритмы	10
6.3	Класс NPR	10
6.4	Декомпозиция f в NPR	10
6.5	Асимптотическая иерархия функций	11
6.6	Подклассы в NPR	11
6.6.1	Подклассы NPR(Low)	11
6.6.2	NPRE (Eq)	11
6.6.3	NPRH	11
6.6.4	TSP	12

Глава 1

Исторический очерк

1. 1900

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

2. 1936

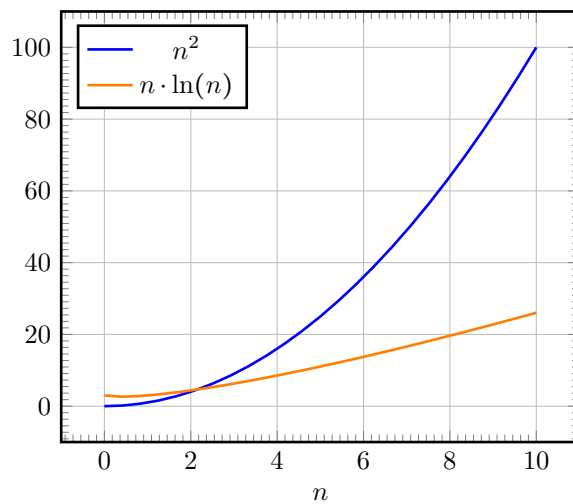
А.Тьюринг, Э.Л.Пост - Теория алгоритмов (начало)

- 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:

А. Новый (метод разработки)

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

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

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

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

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 \rightarrow множество всех конкретных

Решение общей \rightarrow решение каждой конкретной

3.3 Пространство символов

\leftarrow							\rightarrow
--------------	--	--	--	--	--	--	---------------

	Помечен, не помечен	
--	---------------------	--

Конкретная проблема задается внешней силой путем пометки конечного числа символов.

3.4 Работник (процессор)

1. \rightarrow (R)
2. \leftarrow (L)
3. \checkmark 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

	\checkmark	\checkmark	\checkmark	\checkmark	...	\checkmark	
--	--------------	--------------	--------------	--------------	-----	--------------	--

1. ξ
2. \rightarrow

3. $\begin{matrix} \text{да} \\ \rightarrow 1 \\ \text{нет} \\ \rightarrow 4 \end{matrix}$
4. stop

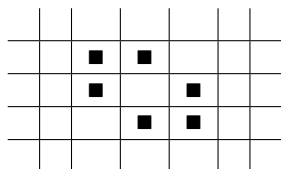
3.6 Гипотеза Поста

- 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



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

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

Глава 4

Терминология

4.1 Вход алгоритма

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

4.2 Длина входа

$n \rightarrow |D|?$

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

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

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

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

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

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

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

4.6 Память

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

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

Глава 5

Примеры

5.1 Умножение матриц

```
1 MultM(A,B,n;C)
2   for i ← 1 to n
3     for j ← 1 to n
4       s ← 0
5       for k ← 1 to n
6         s ← s + A[i,k] * B[k,j]
7       c[i,j] ← s
8 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

```
1 Max(A,n;M)
2   M ← A[1]
3   for j ← 2 to n
4     if M < A[j]
5       M ← A[j]
6 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

```
1 Pow(x,k;y)
2   ...
3   y → 1
4   for j → 1 to k
5     y → y * x
6 End
```

$$D \in D_2$$

$$M_z(D) = 2$$

$$f_A(k) \equiv 2 + 5k$$

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

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

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

Глава 6

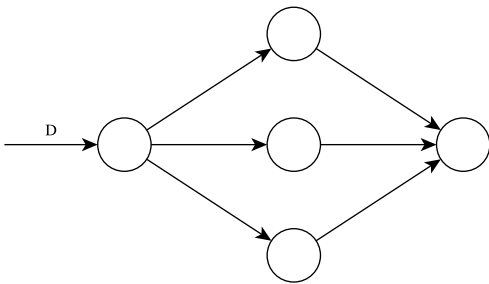
Классификация алгоритмов по типу
трудоемкости

6.1 Класс N - количественно-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

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

$f_A^{\wedge}(n) = f_n(\overline{n}) + g_{PR}^{\wedge}(n)$

6.5 Асимптотическая иерархия функций

$$a < b$$

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

$$f(x) < g(x)$$

$$<: \lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} = 0$$

$$\ln x < \sqrt{x} < x^{\frac{1}{2}} < e^{\lambda x} < x^x$$

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

6.6 Подклассы в NPR

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

6.6.1 Подклассы NPR(Low)

$$g^\wedge(n) < f_n(n)$$

6.6.2 NPRE (Eq)

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

6.6.3 NPRH

$$f_n(n) < g^\wedge(n)$$

$$!g^\wedge \rightarrow \bar{g}$$

6.6.4 TSP

