计算机科学导论

孙晓明

中国科学院计算技术研究所

2022-4-15

问题

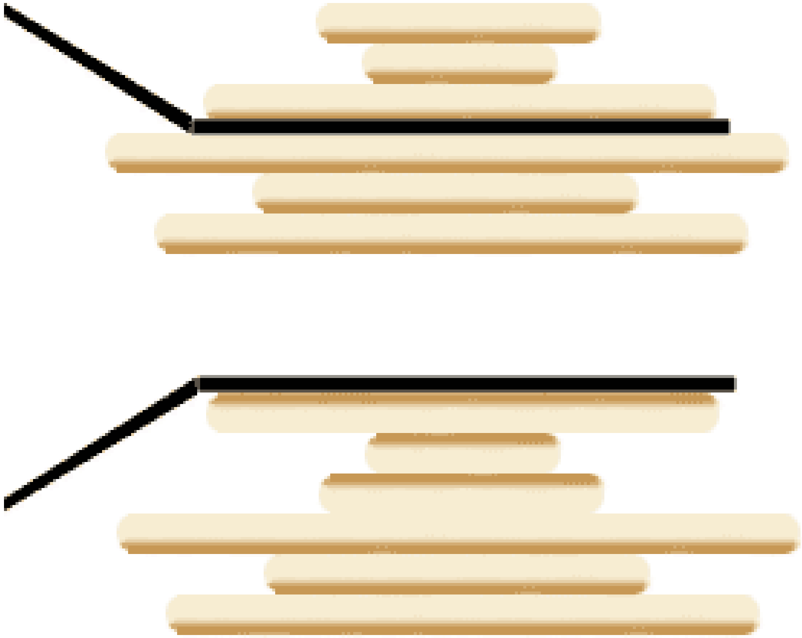
- 使用冒泡排序法对 5,4,3,2,1 进行从小到大排序，需要进行____次交换

- 1) 9
- 2) 10
- 3) 11
- 4) 15

思考题

- 翻煎饼问题(Pancake Sorting)

一名厨师做了一叠大小不同的煎饼，他要不断从上面取几个煎饼进行翻面。假设一共有n张煎饼，厨师需要翻动多少次才能把煎饼按从小到大排好？



翻煎饼问题

- 2, 1, 4, 5, 3
- 5, 4, 1, 2, 3
- 3, 2, 1, 4, 5
- 1, 2, 3, 4, 5

worst case: $2n - 3$

1, 3, 2

2, 4, 3, 1

3, 1, 5, 4, 2

4, 2, 6, 5, 1, 3

……………………..

BOUNDS FOR SORTING BY PREFIX REVERS

William H. GATES

Microsoft, Albuquerque, New Mexico

Christos H. PAPADIMITRIOU*†

Department of Electrical Engineering, University of California, Berkeley,

Received 18 January 1978
Revised 28 August 1978

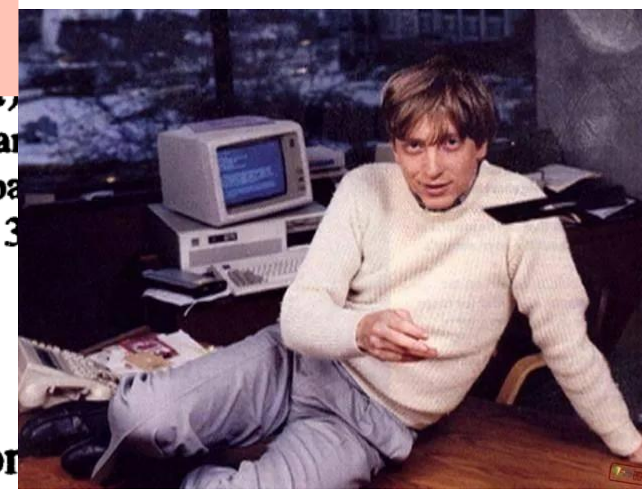
2009: $\frac{18}{11}n + O(1)$. 能否继续改进？

for all σ in (the symmetric group) S_n. We show that $f(n) \leqslant (5n+5)/3$, ar
n a multiple of 16. If, furthermore, each integer is required to participa
reversed prefixes, the corresponding function $g(n)$ is shown to obey 3

1. Introduction

We introduce our problem by the following quotation fror

Christos papadimitriou



Bill Gates

前情回顾

- 单因素优选法
- 冒泡排序
- 归并排序
- 快速排序
- 复杂度：渐进分析

小o，大O记号

- 冒泡排序
  - 运行时间 $O(n^2)$
- 快速排序
  - 平均运行时间 $O(n\log n)$
  - 是$o(n^2)$

小o大O记号 $\{f(n)\}_{n\geq 1}, \{g(n)\}_{n\geq 1}$

- $f(n) = o(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
  - $1000n\log n = o(n^2)$
  - $n^{1000} = o(2^n)$
  - $\log^{200} n = o(n^{0.001})$
- $f(n) = O(g(n))$ if ∃常数$c > 0$,使得 $f(n) \leq cg(n)$ 对充分大的n都成立
  - $1000n\log n = O(n^2)$
  - $\boldsymbol{10^{1000}n = O(n)}$

$\Omega(\cdot), \Theta(\cdot)$记号

- $f(n) = \Omega\big(g(n)\big)$ if $\exists$常数$c > 0$,使得 $f(n) \geq cg(n)$ 对充分大的n都成立
  - $n^2 = \Omega(n \log n)$
  - $0.0001n^3 = \Omega(n^3)$
- $f(n) = \Theta(g(n))$ iff $f(n) = O\big(g(n)\big) \wedge f(n) = \Omega\big(g(n)\big)$
  - $10n^2 - 20n + 45 = \Theta(n^2)$
- 思考: $2^{\Theta(n)}$ 和 $\Theta(2^n)$ 一样吗?

乘法(1)

n^2次乘法运算

- 输入：$X = x_n x_{n-1} \cdots x_1, Y = y_n \cdots y_1$
- 输出：$Z = XY$
- idea:

$$X = X_1 \times 10^{n/2} + X_2$$

$$Y = Y_1 \times 10^{n/2} + Y_2$$

$$Z = XY =$$

$$X_1 Y_1 \times 10^n + (X_1 Y_2 + X_2 Y_1) \times 10^{n/2} + X_2 Y_2$$

```
   123
  ×321
 -----
   123
  246
 369
 -----
 39483
```

乘法(2)

- 分别计算： $X_1Y_1, X_1Y_2, X_2Y_1, X_2Y_2$
- 乘法次数： $T(n) = 4T\left(\frac{n}{2}\right), T(1) = 1$
- $T(n) = n^2 \ldots$

- **计算：** $X_1Y_1, X_2Y_2, (X_1 + X_2)(Y_1 + Y_2)$

$$X_1Y_2 + X_2Y_1 = (X_1 + X_2)(Y_1 + Y_2) - X_1Y_1 - X_2Y_2$$

乘法(3)

- 乘法次数： $T(n) = 3T\left(\frac{n}{2}\right), T(1) = 1$
- $T(n) = n^{\log_2 3} \approx n^{1.59}$
- **总的时间复杂度：**

$$S(n) = 3S\left(\frac{n}{2}\right) + cn$$

$$S(n) = O(n^{\log_2 3})$$

乘法(4)

- 能否更快？
  - 分成3段？

$$X = X_2 \times 10^{2n/3} + X_1 \times 10^{n/3} + X_0$$
$$Y = Y_2 \times 10^{2n/3} + Y_1 \times 10^{n/3} + Y_0$$

$$Z = XY$$
$$= X_2Y_2 \times 10^{4n/3} + (X_1Y_2 + X_2Y_1) \times 10^{n}$$
$$+ (X_0Y_2 + X_1Y_1 + X_2Y_0) \times 10^{2n/3}$$
$$+ (X_1Y_0 + X_0Y_1) \times 10^{n/3} + X_0Y_0$$

乘法(5)

- 方案一：

$$X = X_2 \times 10^{2n/3} + X_1 \times 10^{n/3} + X_0$$
$$Y = Y_2 \times 10^{2n/3} + Y_1 \times 10^{n/3} + Y_0$$

计算X_2Y_2, X_1Y_1, X_0Y_0,

以及$(X_2 + X_1)(Y_2 + Y_1), (X_1 + X_0)(Y_1 + Y_0),$
$(X_2 + X_0)(Y_2 + Y_0)$

$T(n) = 6T(n/3) + cn$ ➡ $T(n) = O(n^{\log_3 6}) \sim n^{1.631}$

方案二：

$$\omega = e^{i\frac{2\pi}{3}}$$

$$A_0 = X_2 + X_1 + X_0, \qquad B_0 = Y_2 + Y_1 + Y_0,$$

$$A_1 = X_2 + \omega X_1 + \omega^2 X_0, \quad B_1 = Y_2 + \omega Y_1 + \omega^2 Y_0,$$

$$A_2 = X_2 + \omega^2 X_1 + \omega X_0, \quad B_2 = Y_2 + \omega^2 Y_1 + \omega Y_0.$$

$$A_0B_0 + A_1B_1 + A_2B_2 = 3(X_2Y_2 + X_1Y_0 + X_0Y_1)$$

$$A_0B_0 + \omega A_1B_1 + \omega^2 A_2B_2 = 3(X_0Y_2 + X_1Y_1 + X_2Y_0)$$

$$A_0B_0 + \omega^2 A_1B_1 + \omega A_2B_2 = 3(X_2Y_1 + X_1Y_2 + X_0Y_0)$$

乘法(6)

- 计算$X_2Y_2, X_0Y_0, A_2B_2, A_1B_1, A_0B_0$
- $T(n) = 5T\left(\frac{n}{3}\right) + cn$

  $T(n) = O\left(n^{\log_3 5}\right) \sim n^{1.465}$

- 思考题：能否更快？
- **快速傅立叶变换(FFT)**
  - idea：看作$f(z)$ $g(z)$，使用拉格朗日插值公式

- $f(z) = x_0 + x_1 z + \ldots + x_n z^n$
- $g(z) = y_0 + y_1 z + \ldots + y_n z^n$
- $h(z) = f(z)\, g(z)$, $h(10)$ or $h(2)$
- 令 $N=2n+2$，取$\alpha_1, \alpha_2, \ldots, \alpha_N$
  - 计算$f(\alpha_1), \ldots, f(\alpha_N), g(\alpha_1), \ldots, g(\alpha_N)$
  - 计算 $h(\alpha_i) = f(\alpha_i)g(\alpha_i)$ $(i = 1, 2, \ldots, N)$
  - 计算$h(z)$

快速傅立叶变换(Fast Fourier Transform)

- 离散傅立叶变换(Discrete Fourier Transform)

$$DFT_n : (x_0, x_1, \ldots, x_{n-1}) \in C^n \to (y_0, y_1, \ldots, y_{n-1}) \in C^n$$

$$y_i = \sum_{j=0}^{n-1} e^{-\frac{2\sqrt{-1}\pi}{n} ji} x_j, (i = 0, 1, \ldots, n-1)$$

$$
\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-\frac{2\sqrt{-1}\pi}{n}} & \cdots & e^{-\frac{2\sqrt{-1}\pi}{n}(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-\frac{2\sqrt{-1}\pi}{n}(n-1)} & \cdots & e^{-\frac{2\sqrt{-1}\pi}{n}(n-1)^2} \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix}
$$

记 $\omega_n = e^{-\frac{2\sqrt{-1}}{n}\pi}$, 假定 $n = 2^k$,

$$y_i = \sum_{j=0}^{n-1} \omega_n^{ij} x_j = \sum_{j_1=0}^{n/2-1} \omega_n^{2ij_1} x_{2j_1} + \sum_{j_2=0}^{n/2-1} \omega_n^{i(2j_2+1)} x_{2j_2+1}$$

$$= \sum_{j=0}^{n/2-1} \omega_{n/2}^{ij} x_{2j} + \omega_n^{i} \sum_{j=0}^{n/2-1} \omega_{n/2}^{ij} x_{2j+1}$$

$$\omega_n^{2j} = \omega_{n/2}^{j}$$

$$y_i = \sum_{j=0}^{n/2-1} \omega_{n/2}^{ij} x_{2j} + \omega_n^i \sum_{j=0}^{n/2-1} \omega_{n/2}^{ij} x_{2j+1} (i = 0,1,\ldots, n/2-1)$$

$$y_{i+n/2} = \sum_{j=0}^{n/2-1} \omega_{n/2}^{ij} x_{2j} - \omega_n^i \sum_{j=0}^{n/2-1}$$

$$W_{n/2} = \begin{bmatrix} \omega_{n/2}^0 & & \\ & \omega_{n/2}^1 & \\ & & \ddots \\ & & & \omega_{n/2}^{n/2-1} \end{bmatrix}$$

$$Y_{up} = DFT_{n/2}(X_{even}) + W_{n/2} \cdot DFT_{n/2}(X_{odd})$$

$$Y_{down} = DFT_{n/2}(X_{even}) - W_{n/2} \cdot DFT_{n/2}(X_{odd})$$

- 基于快速傅里叶变换
- $O(n \log n \log \log n)$ (Schönhage and Strassen, 1971)
- $O(n \log n \, 2^{O(\log^* n)})$ (Fürer, 2007)
- $O(n \log n \, 8^{\log^* n})$ (Harvey and Hoeven, 2016)
- $O(n \log n \, 4^{\log^* n})$ (Harvey and Hoeven, 2019)
- $O(n \log n)$ (Harvey and Hoeven, 2021)
  - $\log^* n \triangleq \min\{k : \overbrace{\log \log \cdots \log n} \leq 1\}$

矩阵乘法(1)

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \ddots & & \\ \vdots & & \ddots & \\ a_{n1} & & & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & \ddots & & \\ \vdots & & \ddots & \\ b_{n1} & & & b_{nn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & \ddots & & \\ \vdots & & \ddots & \\ c_{n1} & & & c_{nn} \end{bmatrix}$$

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj}$$

$O(n^3)$次**乘法**， $O(n^3)$次加法

矩阵乘法(2)

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

$$m_1 = (a_{12} - a_{22})(b_{21} + b_{22})$$

$$m_2 = (a_{11} + a_{22})(b_{11} + b_{22})$$

$$m_3 = (a_{11} - a_{21})(b_{11} + b_{12})$$

$$m_4 = (a_{11} + a_{12})b_{22}$$

$$m_5 = a_{11}(b_{12} - b_{22})$$

$$m_6 = a_{22}(b_{21} - b_{11})$$

$$m_7 = (a_{21} + a_{22})b_{11}$$

$$c_{11} = m_1 + m_2 - m_4 + m_6$$

$$c_{12} = m_4 + m_5$$

$$c_{21} = m_6 + m_7$$

$$c_{22} = m_2 - m_3 + m_5 - m_7$$

$O(n^{\log 7 \approx 2.81})$次**乘法**

矩阵乘法(3)

- Strassen algorithm’69 $O(n^{2.81})$
- $O(n^{2.79}), O(n^{2.55}), O(n^{2.48}) \ldots$
- Coppersmith–Winograd algorithm’89 $O(n^{2.376})$
- Stothers’10 $O(n^{2.374})$
- Williams’11 $O(n^{2.373})$
- Le Gall’14 $O(n^{2.3729})$
- Alman, Williams’21 $O(n^{2.3728596})$

思考题

- 汉诺塔（Hanoi）
- 如果有4根柱子怎么办？

$2^n - 1$

谢谢！