Divide and Conquer

Lecture outline

Pierre Senellart

4 October 2018

Call the roll. Preliminary questions?

1 Brute force

- Exhaustive exploration
- Examples: linear search, sort
- Often possible, always consider before discarding the option
- Sometimes the only solution
- Sometimes very small constant, ok for small input sizes

2 Divide and Conquer (4.1, 4.2)

- Historical origin (*Divide et impera*, attributed to Philip II of Macedon, used in politics, in military strategy)
- In computer science: divide, process recursively, merge
- Often allows (but not always) saving time

2.1 Example: binary search

2.2 Example: matrix multiplication

- Standard algorithm
- Naïve divide and conquer, superficial analysis, recurrence formula
- Mention Strassen algorithm and its recurrence formula

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1; \\ 7T(\frac{n}{2}) + \Theta(n^2) & \text{if } n > 1. \end{cases}$$

3 Analyzing Recursive Algorithms (4.4)

- Simplifying assumption of powers of 2 (or of another number), not critical
- Recurrence trees for binary search, naïve matrix multiplication, Strassen ($\log_2 7 \approx 2.807$)
- Practical use of Strassen algorithm
- Mention of modern matrix multiplications (Coppersmith–Winograd and improvements, 2.375477 (1990), 2.374 (2010), 2.3728642 (2011), 2.3728639 (2014))

4 Master theorem (4.5 et 4.6)

$T(n) = aT(\frac{n}{b}) + f(n)$. Let $c = \log_b a$.

1. If $f(n) = O(n^{c'})$ with $c' < c$: $T(n) = \Theta(n^c)$.
2. If $f(n) = \Theta(n^c \log^k(n))$: $T(n) = \Theta(n^c \log^{k+1}(n))$;
3. If $f(n) = \Omega(n^{c'})$ with $c' > c$ and if $af(\frac{n}{b}) \leq \alpha f(n)$ for $\alpha < 1$ with n large enough: $T(n) = \Theta(f(n))$.

- Not a trichotomy
- Examples of uses
- One still assumes n is a power of b
- Generalized by Akra–Bazzi (non-equal input size)
- Proof! Essentially, Lemma 4.3 of textbook

5 Fast Polynomial Multiplication, Discrete Fourier Transform (30)

- Problem
- Naïve method, $O(n^2)$
- Representation as sets of points, proof scheme
- nth roots of 1
- Discrete Fourier transform (DFT) – connection with signal processing
- Fast Fourier transform: decomposition in even/odd components, DFT from these two decompositions, high-level algorithm scheme
- Complexity analysis with the master theorem
- Inverse DFT, Vandermonde matrix inverse, reuse of fast Fourier transform