Διαίρει-και-Βασίλευε

Δημήτρης Φωτάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Πανεπιστήμιο Αιγαίου

Διαίρει-και-Βασίλευε

- Γενική μέθοδος σχεδιασμού αλγορίθμων:
  - **Διαίρεση** σε (≥ 2) υπο-προβλήματα (σημαντικά) μικρότερου μεγέθους.
  - **Ανεξάρτητη** (αναδρομική) επίλυση υπο-προβλημάτων (για μικρά υπο-προβλήματα εφαρμόζουμε στοιχειώδεις αλγορίθμους).
  - **Σύνθεση** λύσης αρχικού προβλήματος από λύσεις υπο-προβλημάτων.
- Ισχυρή μέθοδος, με πολλές σημαντικές εφαρμογές !
- (Εύκολη) ανάλυση με **αναδρομικές εξισώσεις.**
- **Ταξινόμηση :** merge-sort, quicksort.
- **Επιλογή :** quickselect.

Προϋποθέσεις Εφαρμογής

- Διαίρεση σημαντικά ευκολότερη από επίλυση αρχικού.
- Υπο-στιγμιότυπα σημαντικά μικρότερα από αρχικό (π.χ. αρχικό μέγεθος *n*, υπο-στιγμ. μεγέθους *n* / *c*, *c* > 1).
- Ανεξάρτητα υπο-στιγμιότυπα που λύνονται από ανεξάρτητες αναδρομικές κλήσεις.
  - Ίδια ή επικαλυπτόμενα υπο-στιγμιότυπα : σημαντική αύξηση χρόνου εκτέλεσης.
  - Επικαλυπτόμενα υπο-στιγμιότυπα : **Δυναμικός Προγραμματισμός**
- Σύνθεση σημαντικά ευκολότερη από επίλυση αρχικού.

(Αντι)παράδειγμα

- Υπολογισμός *n*-οστού όρου ακολουθίας Fibonacci.

  $f_n = f_{n-1} + f_{n-2}, n \geq 2$

  $f_0 = 0, f_1 = 1$

```
long fibRec(long n) {
  if (n <= 1) return(n);
  return(fibRec(n-1) + fibRec(n-2)); }
```

- Χρόνος εκτέλεσης:

  $$T(n) = \Theta(1) + T(n-1) + T(n-2)\,,\;\; T(1) = \Theta(1)$$

- Λύση: $T(n) = \Theta(\varphi^n)\,,\; \varphi = \frac{1+\sqrt{5}}{2} \approx 1.618$
- Επικαλυπτόμενα στιγμ.: Εκθετικός χρόνος!
- Αλγόριθμος γραμμικού χρόνου;
- Καλύτερος αλγόριθμος;

```
fib(n) {
  int cur = 1, prev = 0;
  for (i = 2; i <= n; i++) {
      cur = cur + prev;
      prev = cur - prev; }
  return(cur);  }
```

Πολλαπλασιασμός

- Υπολογισμός αθροίσματος x + y , x και y αριθμοί *n*-bits.
  - Κλασσικός αλγόριθμος πρόσθεσης, χρόνος Θ(n).
- Υπολογισμός γινομένου x × y , x και y αριθμοί με *n*-bits.
  - Κλασσικός αλγόριθμος πολ/μού, χρόνος $\Theta(n^2)$.
  - Καλύτερος αλγόριθμος;
- Διαίρει-και-Βασίλευε:
  - Διαίρεση: $x = 2^{n/2} x_h + x_l\,,\; y = 2^{n/2} y_h + y_l$

$$x \times y = 2^n \overbrace{x_h y_h}^{z_h} + 2^{n/2} \overbrace{(x_h y_l + x_l y_h)}^{z_m} + \overbrace{x_l y_l}^{z_l} = 2^n z_h + 2^{n/2} z_m + z_l$$

  - 4 πολλαπλασιασμοί (n / 2)-bits, 2 ολισθήσεις, 3 προσθέσεις.
  - Χρόνος: $T_1(n) = 4\,T_1(n/2) + \Theta(n) \Rightarrow T_1(n) = \Theta(n^2)$

Πολλαπλασιασμός

$$x \times y = 2^n \overbrace{x_h y_h}^{z_h} + 2^{n/2} \overbrace{(x_h y_l + x_l y_h)}^{z_m} + \overbrace{x_l y_l}^{z_l} = 2^n z_h + 2^{n/2} z_m + z_l$$

- Όμως z_m υπολογίζεται με 1 μόνο πολ/μο (n/2+1)-bits.

$$z_m = (x_h + x_l)(y_h + y_l) - x_h y_h - x_l y_l$$

  - 3 πολλαπλασιασμοί (n / 2)-bits, 2 ολισθήσεις, 6 προσθέσεις.
  - Χρόνος: $T(n) = 3\,T(n/2) + \Theta(n) \Rightarrow T(n) = \Theta(n^{\log_2 3})$
- Παράδειγμα: 2576 × 7935 = 20440560

$x_h = 25\,,\; x_l = 76\,,\qquad y_h = 79\,,\; y_l = 35$
$z_h = 25 \times 79 = 1975\,,\quad z_l = 76 \times 35 = 2660$
$z_m = (25 + 76)(79 + 35) - 1975 - 2660 =$
$\quad = 101 \times 114 - 1975 - 2660 = 11514 - 1975 - 2660 = 6879$
$x \times y = 1975 \cdot 10^4 + 6879 \cdot 10^2 + 2660 = 20404560$

Πολλαπλασιασμός Πινάκων

- Υπολογισμός γινομένου C = A × B.
  A, B τετραγωνικοί πίνακες n × n.
- Εφαρμογή ορισμού: $C[i,j] = \sum_{k=1}^{n} A[i,k]B[k,j]$
  - Χρόνος $\Theta(n^3)$ (n^2 στοιχεία, χρόνος Θ(n) για καθένα).
- Διαίρει-και-Βασίλευε:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},\; B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix},\; C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

$$C_{11} = A_{11}B_{11} + A_{12}B_{21}$$
$$C_{12} = A_{11}B_{12} + A_{12}B_{22}$$
$$C_{21} = A_{21}B_{11} + A_{22}B_{21}$$
$$C_{22} = A_{21}B_{12} + A_{22}B_{22}$$

  - 8 πολ/μοι και 4 προσθέσεις πινάκων $\frac{n}{2} \times \frac{n}{2}$
  - Χρόνος: $T_1(n) = 8\,T_1(n/2) + \Theta(n^2) \Rightarrow T_1(n) = \Theta(n^3)$

Αλγόριθμος Strassen (1960)

$$D_1 = (A_{21} + A_{22} - A_{11})(B_{22} - B_{12} + B_{11})$$
$$D_2 = A_{11}B_{11}$$
$$D_3 = A_{12}B_{21}$$
$$D_4 = (A_{11} - A_{21})(B_{22} - B_{12})$$
$$D_5 = (A_{21} + A_{22})(B_{12} - B_{11})$$
$$D_6 = (A_{12} - A_{21} + A_{11} - A_{22})B_{22}$$
$$D_7 = A_{22}(B_{11} + B_{22} - B_{12} - B_{21})$$

$$C_{11} = D_2 + D_3$$
$$C_{12} = D_1 + D_2 + D_5 + D_6$$
$$C_{21} = D_1 + D_2 + D_4 - D_7$$
$$C_{22} = D_1 + D_2 + D_4 + D_5$$

- 7 πολ/μοι και 24 προσθέσεις πινάκων $\frac{n}{2} \times \frac{n}{2}$
  - Χρόνος: $T(n) = 7\,T(n/2) + \Theta(n^2) \Rightarrow T(n) = \Theta(n^{\log_2 7})$

Υπολογισμός Δύναμης (Diffie-Hellman)

- Συμφωνία Αλίκης και Βασίλη σε κρυπτογραφικό κλειδί. Εύα παρακολουθεί για να «κλέψει» το κλειδί.
- A, B συμφωνούν δημόσια σε πρώτο p και ακέραιο $q < p$. E γνωρίζει p, q.
  - Εμπλεκόμενοι αριθμοί είναι πολυψήφιοι (π.χ. 512 ψηφία).
- A διαλέγει τυχαία $a < p$ και υπολογίζει $q_a = q^a \bmod p$
  B διαλέγει τυχαία $b < p$ και υπολογίζει $q_b = q^b \bmod p$
  A, B ανταλλάσσουν q_a, q_b και τα μαθαίνει E.
- A, B υπολογίζουν K (μόνοι τους). E δεν ξέρει K.
  $$K = q_b^a \bmod p = (q^b \bmod p)^a \bmod p = q^{ab} \bmod p$$
- Για K, E χρειάζεται a, b (δεν μεταδόθηκαν). Επίλυση διακριτού λογαρίθμου (πολύ δύσκολο).

Υπολογισμός Δύναμης

- Εφαρμογή υποθέτει **αποδοτικό** αλγόριθμο υπολογισμού $\exp(x, n, p) = x^n \bmod p$, x, n, p πολυψήφιοι ακέραιοι.
  - Υπολογισμός δυνάμεων με τη σειρά (1, 2, 3, …): αν μήκος 512 bits, χρειάζεται περίπου 2^{512} πολ/μους!!!
- Διαίρει-και-Βασίλευε (έστω n άρτιος):
  - Υπολογίζουμε αναδρομικά $\exp(x, n/2, p) = x^{n/2} \bmod p$
  - … και $\exp(x, n, p) = \exp(x, n/2, p) \times \exp(x, n/2, p)$
- Χρόνος:
  $T(n) = T(n/2) + \mathrm{O}(\log^2 p)$
  $\Rightarrow T(n) = \mathrm{O}(\log n \log^2 p)$
  - Μήκος 512 bits: περίπου 2^{10} πολ/μους.

ExponRec(x, n, p)
  if $n = 1$ **then return**$(x \bmod p)$;
  $t \leftarrow$ ExponRec$(x, \lfloor n/2 \rfloor, p)$;
  $t \leftarrow t^2 \bmod p$;
  if n is odd **then return**$(t \times x \bmod p)$;
  else return(t);

Ακολουθία Fibonacci

- Ακολουθία Fibonacci: $f_n = f_{n-1} + f_{n-2}, n \geq 2$, $f_0 = 0, f_1 = 1$
- Θεωρώ πίνακα $A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ και $F_n = [f_n, f_{n-1}]$
  - Παρατηρώ ότι $A \times F_n = [f_n + f_{n-1}, f_n] = F_{n+1}$
  - Με επαγωγή αποδεικνύω ότι $F_n = A^{n-1} \times F_1$, $F_1 = [1, 0]$
- Διαίρει-και-Βασίλευε:
  - Υπολογισμός A^n σε χρόνο O(log n) (όπως με αριθμούς).
  - Υπολογίζω αναδρομικά το $A^{n/2}$ και $A^n = A^{n/2} \times A^{n/2}$
  - Χρόνος: $T(n) = T(n/2) + \Theta(1) \Rightarrow T(n) = \Theta(\log n)$