Αποδοτικοί Αλγόριθμοι

Μέρος 1ο: Εισαγωγή

Εξάμηνο: 7ο

Κωδικός μαθήματος: 748

Τμήμα Μαθηματικών
Πανεπιστήμιο Ιωαννίνων

Μιχάλης Α. Μπέκος
bekos@uoi.gr

Οργάνωση μαθήματος

- Περιεχόμενο μαθήματος
  - Εισαγωγή - Επανάληψη
  - Επιμερισμένη ανάλυση
  - Ελάχιστα διασυνδετικά δέντρα
  - Ελάχιστες τομές
  - Μέγιστες ροές
  - Επίπεδα γραφήματα
  - Προσεγγιστικοί αλγόριθμοι
  - Τυχαιοκρατικοί αλγόριθμοι
- Εργασίες: Θα υπάρξουν συνολικά δύο φύλλα εργασιών

Εισαγωγή

- Ανάλυση πολυπλοκότητας
- Κάτω όρια
- Μέση / επιμερισμένη ανάλυση

δυσεπίλυτα προβλήματα	$\mathcal{NP}$-πλήρη προβλήματα	
υπολογίσιμα προβλήματα	$\mathcal{P}$	
		γραμμικός προγραμματισμός
	n^3	
		πολλαπλασιασμός πινάκων
	n^2	
	$n \log n$	ταξινόμηση
	n	επιλογή
	$\log n$	δυαδική αναζήτηση

Ανάλυση πολυπλοκότητας: Χρόνος

- Παράδειγμα: Ταξινόμηση με συγκρίσεις

```
BubbleSort(A[1 … n])
{
   for i=1 to n do
   {
      for j=1 to n-i do
      {
         if A[j] > A[j+1]
            swap(A[j], A[j+1]);
      }
   }
}
```

- Χρονική πολυπλοκότητα: $O(n^2)$ (αντιστοιχεί στο πλήθος των συγκρίσεων)
- Απόδειξη ορθότητας: Μετά το βήμα i, ο υποπίνακας $A[n-i \ldots n]$ είναι ταξινομημένος

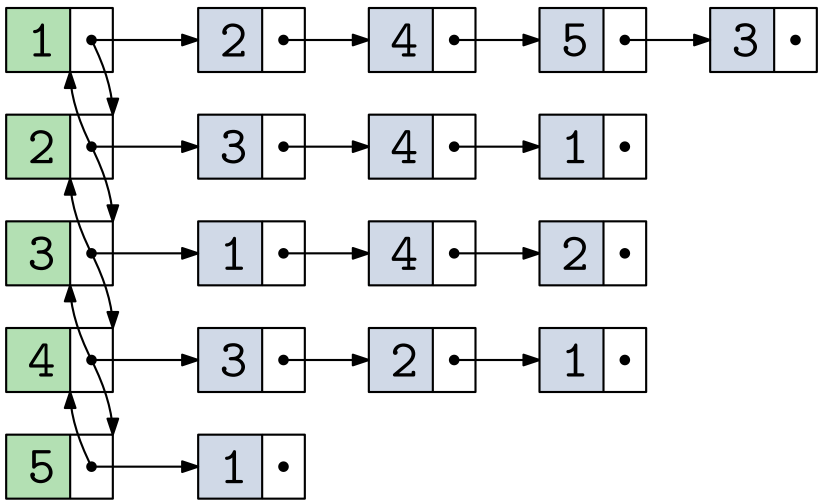
Ανάλυση πολυπλοκότητας: Χώρος

- Παράδειγμα: Αναπαράσταση γραφήματος

$$G = (V, E);\ V = \{1, 2, 3, 4, 5\}$$

$$E = \{(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (3, 4)\}$$

- Λίστα γειτνίασης $O(n + m)$



- Πίνακας $O(n^2)$

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Συμβολισμοί Big-O

- $O()$: Άνω φράγμα
- $\Omega()$: Κάτω φράγμα
- $\Theta()$: Σφιχτό φράγμα
- $o()$: Αυστηρό άνω φράγμα
- $\omega()$: Αυστηρό κάτω γράγμα

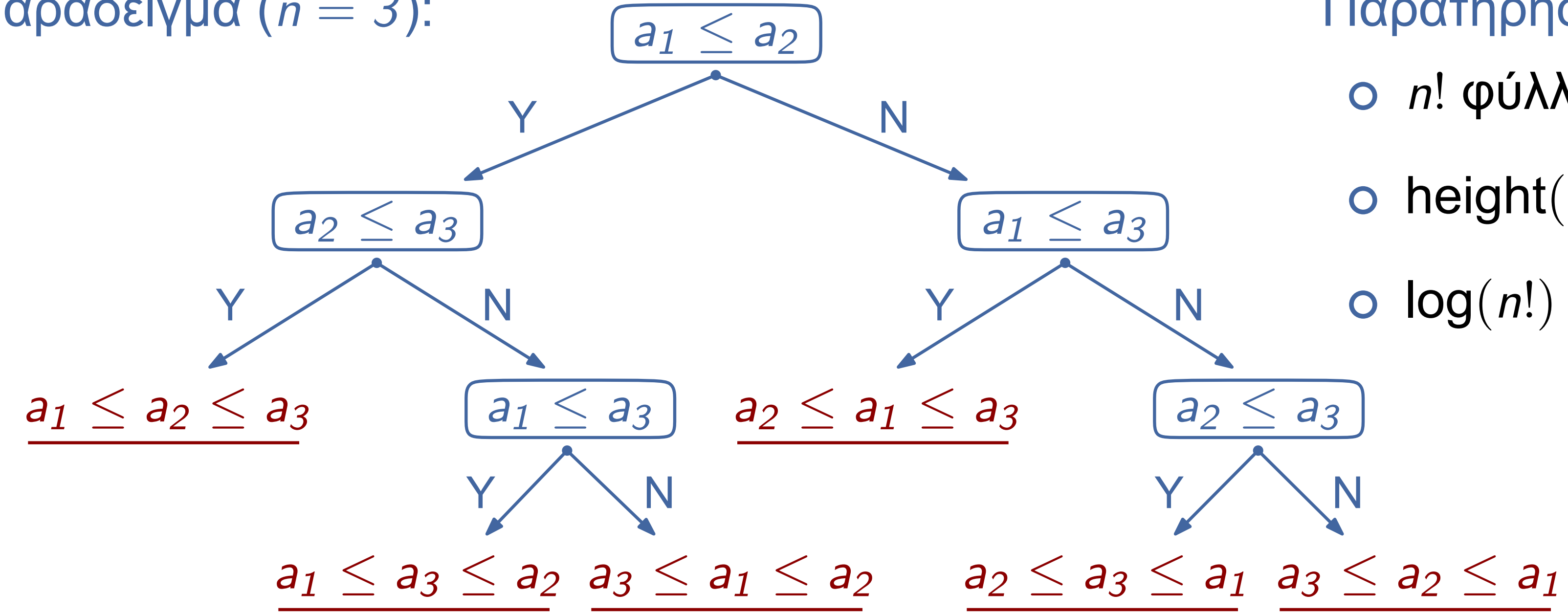
Κάτω φράγματα

- Θεώρημα: Η ταξινόμηση (με συγκρίσεις) είναι $\Omega(n \log n)$

  Απόδειξη:

  Μπορούμε να αναπαραστήσουμε κάθε αλγόριθμο ταξινόμησης (έστω σε αύξουσα σειρά) ως ένα δέντρο απόφασης.

  Παράδειγμα ($n = 3$):



  Παρατηρήσεις:

  - $n!$ φύλλα $\Rightarrow \text{height}(T) \geq \log(n!)$
  - $\text{height}(T) = \#$ συγκρίσεων
  - $\log(n!) = \Theta(n \log n)$

Ταξινόμηση: Συνέχεια

- Θεώρημα: Η ταξινόμηση (με συγκρίσεις) είναι $\Theta(n \log n)$

  Απόδειξη:

```
MergeSort(A, l, r)
{
   if(l < r)
   {
      m = (l+r)/2
      A1 = MergeSort(A, l, m);
      A2 = MergeSort(A, m+1, r);
      return Merge(A1, A2);
   }
   return A[l];
}
```

- Ανάλυση πολυπλοκότητας:

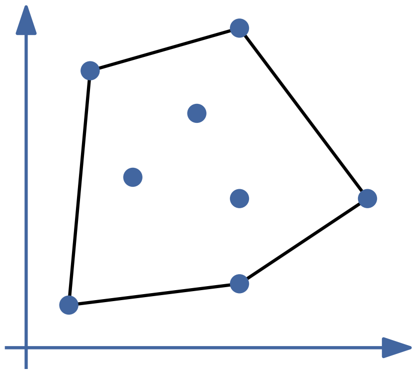
$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + cn$$

$$\Big\downarrow \; n = 2^k \Rightarrow k = \log n$$

$$\begin{aligned}
T(n) &= 2T(n/2) + cn \\
&= 2(2T(n/4) + cn/2) + cn \\
&= 2^2 T(n/2^2) + 2cn \\
&= \cdots \\
&= 2^k T(n/2^k) + kcn \\
&= nT(1) + cn\log n \\
&= O(n\log n)
\end{aligned}$$

- Ερώτημα: Πώς θα αποδεικνύατε την ορθότητα;

Το πρόβλημα του κυρτού περιγράμματος



- Είσοδος: n σημεία $p_1 = (x_1, y_1), \ldots, p_n = (x_n, y_n)$ στο $\mathbb{R}^2$
- Έξοδος: Ένα κυρτό πολύγωνο ελάχιστου εμβαδού που περιέχει τα $p_1, \ldots, p_n$
- Σημείωση: Υπάρχουν διάφοροι αλγόριθμοι για τον υπολογισμό του κυρτού περιγράμματος n σημείων σε χρόνο $\Theta(n \log n)$
- Ερώτημα: Μπορούμε να τα καταφέρουμε καλύτερα από $O(n \log n)$;
- Απάντηση: Όχι

Απόδειξη:

Στιγμιότυπο ταξινόμησης: $a_1, \ldots, a_n \longrightarrow$ Στιγμιότυπο κυρτού περιγράμματος: $(a_1, a_1^2), \ldots, (a_n, a_n^2)$

Παρατήρηση: Όλα τα σημεία βρίσκονται στο κυρτό περίγραμμα $\Rightarrow$

Αν υπάρχει αλγόριθμος για τον υπολογισμό του κυρτού περιγράμματος σε $o(n \log n)$ χρόνο, τότε η ταξινόμηση μπορεί να γίνει σε $o(n \log n)$ χρόνο (άτοπο).

Χρήσιμες ιδιότητες

- Ιδιότητες λογαρίθμων:
  - $\log_x a^b = b \cdot \log_x a$
  - $\log_x ab = \log_x a + \log_x b$
  - $\log_x \frac{a}{b} = \log_x a - \log_x b$
  - $\log_x a = \log_y a / \log_y x$
  - $x = y^{\log_y x}$
- Ιδιότητες αθροισμάτων:
  - $\sum_{i=1}^{\infty} q^i = \frac{1}{1-q}$; για $|q| < 1$
  - $\sum_{i=1}^{n} q^i = \frac{q^{n+1}-1}{q-1}$
  - $\sum_{i=1}^{n} \frac{1}{i} = H_n$; όπου $\ln n \leq H_n \leq \ln n + 1$

Ανάλυση χρονικής πολυπλοκότητας αλγορίθμων

- Παράδειγμα 1: $T(n) = 2T(n) + \log n \qquad (1)$

Trick 1: Θέτω $m = \log n \iff n = 2^m$

Τότε: (1) $\iff T(2^m) = 2T(2^{m/2}) + m \qquad (2)$

Trick 2: Θέτω $S(m) = T(2^m)$

Τότε: (2) $\iff S(m) = 2S(m/2) + m = O(m \log m)$

$\Rightarrow T(n) = T(2^m) = S(m) = O(\log n \log\log n)$

- Παράδειγμα 2: $T(n) = 3T(n/4) + n$

Διαίρει και βασίλευε: Πολλαπλασιασμός πινάκων

- Είσοδος: Δύο $n \times n$ πίνακες A και B
- Έξοδος: Ένας $n \times n$ πίνακας C έτσι ώστε $C = A \cdot B$

- Μια ιδέα διαίρει και βασίλευε:

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \cdot \begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix} = \begin{bmatrix} A_1B_1 + A_2B_3 & A_1B_2 + A_2B_4 \\ A_3B_1 + A_4B_3 & A_3B_2 + A_4B_4 \end{bmatrix} = \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix}$$

Χώρισε τους A και B σε 4 μέρη

8 πολλαπλασιασμοί & 4 προσθέσεις

- Χρονική πολυπλοκότητα: $T(n) = 8T(n/2) + cn^2 = \ldots = O(n^3)$

Διαίρει και βασίλευε: Το σχήμα του Strassen

- $C_1 = P_5 + P_4 - P_2 + P_6$
- $C_2 = P_1 + P_2$
- $C_3 = P_3 + P_4$
- $C_4 = P_5 + P_1 - P_3 - P_7$

- $P_1 = A_1(B_2 - B_4)$
- $P_2 = (A_1 + A_2)B_4$
- $P_3 = (A_3 + A_4)B_1$
- $P_4 = A_4(B_3 - B_1)$
- $P_5 = (A_1 + A_4)(B_1 + B_4)$
- $P_6 = (A_2 - A_4)(B_3 + B_4)$
- $P_7 = (A_1 - A_3)(B_1 + B_2)$

- Παρατήρηση: 7 πολλαπλασιασμοί & 18 προσθέσεις
- Χρονική πολυπλοκότητα: $T(n) = 7T(n/2) + cn^2 = \ldots = O(n^{\log 7}) = O(n^{2.807})$

Ανάλυση πολυπλοκότητας μέσης περίπτωσης

- Είσοδος: Ένας μη ταξινομημένος πίνακας $A[1 \ldots n]$
- Έξοδος: Το ελάχιστο στοιχείο του A

```
FindMin(A[1 ... n])
{
   min = A[1];            ──► Κόστος c1
   for i=2 to n do
   {
      if A[i] < min       ──► Κόστος c2
         min = A[i];      ──► Κόστος c3
   }
}
```

Αυτή η ανάθεση θα κληθεί $n-1$ φορές;
Πολύ απίθανο!

- Κόστος χειρότερης περίπτωσης: $c_1 + (n-1)c_2 + (n-1)c_3$

Ανάλυση πολυπλοκότητας μέσης περίπτωσης

- Μέση περίπτωση: Έστω X_i τυχαία διακριτή μεταβλητή τέτοια ώστε
  - $X_i = 1$, εάν $A[i] < A[1], \ldots, A[i-1]$
  - $X_i = 0$, διαφορετικά
- Η X_i είναι $0/1 \Rightarrow E[X_i] = Pr(X_i = 1) = 1/i$
- Συνεπώς, το αναμενόμενο κόστος του αλγορίθμου είναι:

$$c_1 + (n-1)c_2 + c_3 E[X_2 + \ldots + X_n] =$$
$$c_1 + (n-1)c_2 + c_3(E[X_2] + \ldots + E[X_n]) =$$
$$c_1 + (n-1)c_2 + c_3(1/2 + \ldots + 1/n) \leq$$
$$c_1 + (n-1)c_2 + c_3(\ln n + 1)$$

Λιγότερο από $n-1$ όπως αναμενόταν

Επιπλέον υλικό

- Κεφάλαιο 1, ενότητα 2.1 από τις σημειώσεις του μαθήματος.
  Διαθέσιμες στη σελίδα του μαθήματος στο ecourse.