

Επιλογή

Δημήτρης Φωτάκης

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Εθνικό Μετσόβιο Πολυτεχνείο



Πρόβλημα Επιλογής

- Πίνακας $A[]$ με n στοιχεία (όχι ταξινομημένος).
Αριθμός k , $1 \leq k \leq n$.
- Υπολογισμός του k -οστού μικρότερου στοιχείου
(στοιχείο θέσης $A[k]$ αν A ταξινομημένος).
 - $k = 1$: ελάχιστο. $k = n$: μέγιστο.
 $k = n / 2$: ενδιάμεσο (median).

5	3	2	6	4	1	3	7
---	---	---	---	---	---	---	---

Ελάχιστο : **1**

Μέγιστο : **7**

Ενδιάμεσο : **3**

Εφαρμογές

- Υπολογισμός **στατιστικού ενδιάμεσου** (median).
 - Χρήσιμες πληροφορίες για κατανομή.
 - Ανήκει η Ελλάδα στο φτωχότερο 25% των χωρών ΕΕ;
 - Ανήκει κάποιος φοιτητής στο καλύτερο 10% του έτους του;
- **Ισομερής διαίρεση** (partition) πίνακα σε **ομάδες «ταξινομημένες»** μεταξύ τους.
- Ενδιαφέρον **αλγοριθμικό πρόβλημα!**

Μέγιστο / Ελάχιστο

- Μέγιστο (ελάχιστο) εύκολα σε χρόνο $\Theta(n)$, με $n - 1$ συγκρίσεις μεταξύ στοιχείων.

```
int maximum(int A[], int n) {  
    int max = A[0], i;  
    for (i = 1; i < n; i++)  
        if (A[i] > max) max = A[i];  
    return(max);    }
```

- Μέγιστο και ελάχιστο με $3 \lfloor n/2 \rfloor$ συγκρίσεις ! Πώς;

Κάτω Φράγμα για Μέγιστο

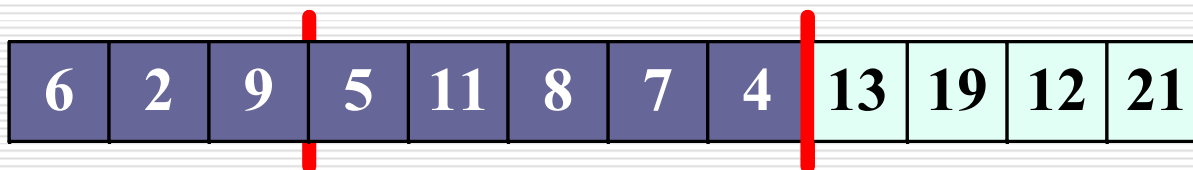
- Κάθε ντετερμινιστικός συγκριτικός αλγόριθμος χρειάζεται $\geq n - 1$ συγκρίσεις για μέγιστο (ελάχιστο).
 - Κάθε σύγκριση αποκλείει την περίπτωση ένα από τα δύο εμπλεκόμενα στοιχεία να είναι μέγιστο.
 - «Πρωτάθλημα» μεταξύ στοιχείων.
 - Σύγκριση στοιχείων : αγώνας όπου κερδίζει μεγαλύτερο.
 - Κάθε «αήττητο» στοιχείο είναι υποψήφιο μέγιστο.
 - Για μοναδικό μέγιστο, πρέπει τα υπόλοιπα να «ηττηθούν».
 - Κάθε αγώνας δίνει ένα «ηττημένο» στοιχείο
 - $\geq n - 1$ αγώνες / συγκρίσεις για μοναδικό μέγιστο.

Επιλογή

- Σε χρόνο $O(n \log n)$ με ταξινόμηση.
- Μέγιστο ($k = 1$), ελάχιστο ($k = n$) : χρόνος $\Theta(n)$.
- Άλλες τιμές k : χρόνος $O(n \log n)$ ή $O(n)$;
- Επιλογή σε γραμμικό χρόνο με διαίρει-και-βασίλευε βασισμένη σε διαχωρισμό της quicksort !

Πιθανοτική Quickselect

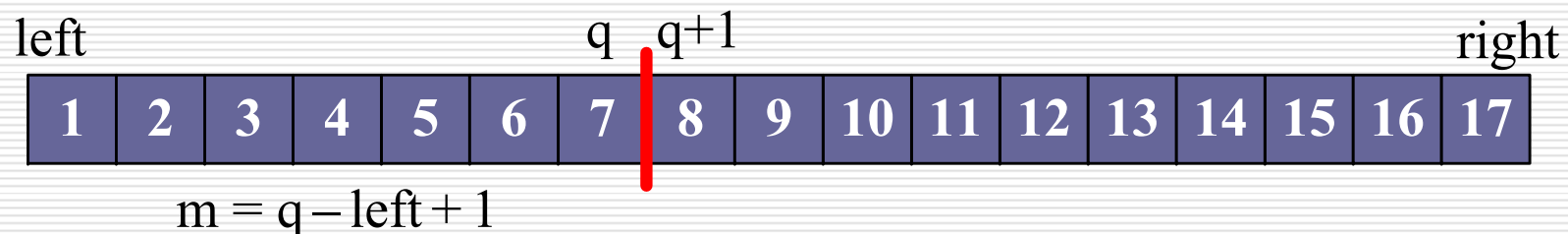
- Έστω υπο-πίνακας $A[l...r]$ και αναζητούμε k -οστό στοιχείο.
- **Τυχαίο** στοιχείο διαχωρισμού (pivot).
- **Αναδιάταξη** και **διαίρεση** εισόδου σε δύο υπο-ακολουθίες:
 - Στοιχεία αριστερής $[l...q]$ υπο-ακολ. \leq στοιχείο διαχωρισμού.
 - Στοιχεία δεξιάς $[q+1...r]$ υπο-ακολ. \geq στοιχείο διαχωρισμού.
- Αν $k \leq q-l+1$, αναδρομική λύση $(A[l...q], k)$
Αν $k > q-l+1$, αναδρομική λύση $(A[q+1...r], k-(q-l+1))$



$$k = 6$$

Ορθότητα Quickselect

- Τερματισμός : μέγεθος υπο-ακολουθιών $\leq n - 1$.
- Επαγωγικά υποθέτω ότι $1 \leq k \leq \text{right} - \text{left} + 1$.
 - Πλήθος στοιχείων στα αριστερά: $m = q - \text{left} + 1$.
 - Αν $k \leq m$, δεξιά στοιχεία «αποκλείονται».
 - Αν $m < k$, αριστερά στοιχεία «αποκλείονται» και k μειώνεται αντίστοιχα ($k' = k - m$).



Πιθανοτική Quickselect

```
int RQuickSelect(int A[], int left, int right, int k)
{
    if (left == right) return(A[left]); // 1 στοιχείο

    pivot = random(left, right); // τυχαίο pivot
    swap(A[left], A[pivot]);

    q = partition(A, left, right); // διαίρεση
    nel = q - left + 1; // #στοιχείων στο αριστερό μέρος

    if (k <= nel) return(RQuickSelect(A, left, q, k));
    else return(RQuickSelect(A, q+1, right, k - nel));
}
```

Χρόνος Εκτέλεσης (χ.π.)

- Χρόνος εκτελ. **αναδρομικών αλγ.** με διατύπωση και λύση **αναδρομικής εξίσωσης** λειτουργίας.
- **$T(n)$** : χρόνος (χ.π.) για **επιλογή από n στοιχεία**.
- Χρόνος εκτέλεσης **partition**(n στοιχεία) : $\Theta(n)$
- Χειρότερη περίπτωση : **ένα στοιχείο «αποκλείεται»** σε κάθε διαίρεση!

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

$$T(n) = \Theta(n) + \Theta(n - 1) + \Theta(n - 2) + \dots + \Theta(1) = \Theta(n^2)$$

- **Πιθανοτικός αλγ.:** χειρότερη περίπτωση έχει **εξαιρετικά μικρή πιθανότητα** να συμβεί (για κάθε είσοδο) !

Χρόνος Εκτέλεσης (μ.π.)

- **Καλή περίπτωση** : διαίρεση $(n/4, 3n/4)$ ή καλύτερη.
 - Τουλάχιστον $n/4$ στοιχεία «αποκλείονται».
- Πιθανότητα «καλής περίπτωσης» $\geq 1/2!$
 - Κατά «μέσο όρο», μία «κακή διαίρεση» πριν από «καλή διαίρεση» που μειώνει στοιχεία από n σε $\leq 3n/4$.

$$S(n) = \Theta(n) + S(3n/4)$$

- Λύση αναδρομής: $S(n) = \Theta(n)$
 - Γεωμετρική σειρά :

$$S(n) \leq cn + \frac{3}{4}cn + \left(\frac{3}{4}\right)^2 cn + \left(\frac{3}{4}\right)^3 cn + \dots + c = \Theta(n)$$



Χρόνος Εκτέλεσης (μ.π.)

□ Τυχαίο στοιχείο σαν στοιχείο χωρισμού (pivot).

□ Για κάθε $i \in [n - 1]$,

πιθανότητα διαίρεσης $(i, n - i) = \frac{1}{n - 1}$

$$\begin{aligned} S(n) &= \Theta(n) + \frac{1}{n - 1} \sum_{i=1}^{n-1} S(\max\{i, n - i\}) \\ &= \Theta(n) + \frac{2}{n - 1} \sum_{i=n/2}^{n-1} S(i) \end{aligned}$$

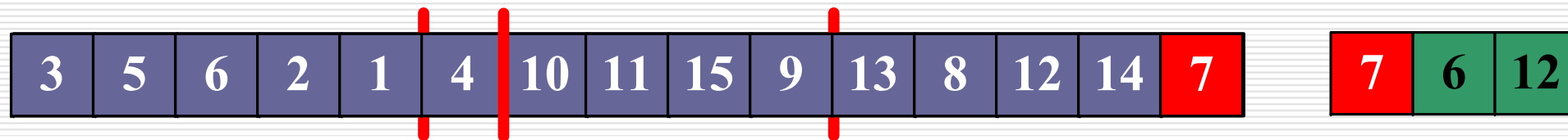
□ Λύση αναδρομής : $S(n) = \Theta(n)$

Ντετερμινιστική Επιλογή

- «Καλή διαίρεση» **ντετερμινιστικά**:
 - Χρήση ρινοτ **κοντά** στο ενδιάμεσο: πρόβλημα επιλογής!
 - Φαύλος κύκλος : γρήγορη επιλογή → καλή διαίρεση → γρήγορη επιλογή.
- Προσεγγιστική επιλογή : όχι «ενδιάμεσο» αλλά «κοντά στο ενδιάμεσο» για ρινοτ.
 - Επιλογή **κατάλληλου** δείγματος (π.χ. $n / 5$ στοιχεία).
 - Ενδιάμεσο δείγματος είναι «κοντά στο ενδιάμεσο» για σύνολο στοιχείων.
 - Αναδρομικά **ενδιάμεσο** στοιχείο του δείγματος.
 - Ενδιάμεσο δείγματος για ρινοτ εγγυάται «καλή διαίρεση».

Ντετερμινιστική Επιλογή

- **Δείγμα:** Χωρίζουμε στοιχεία σε 5άδες.
Βρίσκουμε ενδιάμεσο κάθε 5άδας: $n / 5$ στοιχεία.
 - Χρόνος : $\Theta(n)$.
- Αναδρομικά, ενδιάμεσο στοιχείο δείγματος.
 - Χρόνος : $T(n / 5)$
- Διαίρεση με ενδιάμεσο δείγματος σαν pivot.
 - Χρόνος : $\Theta(n)$.
 - Μεγαλύτερος υποπίνακας έχει $\leq 7n/10$ στοιχεία.
- Αναδρομική επιλογή: **χρόνος $T(7n / 10)$**

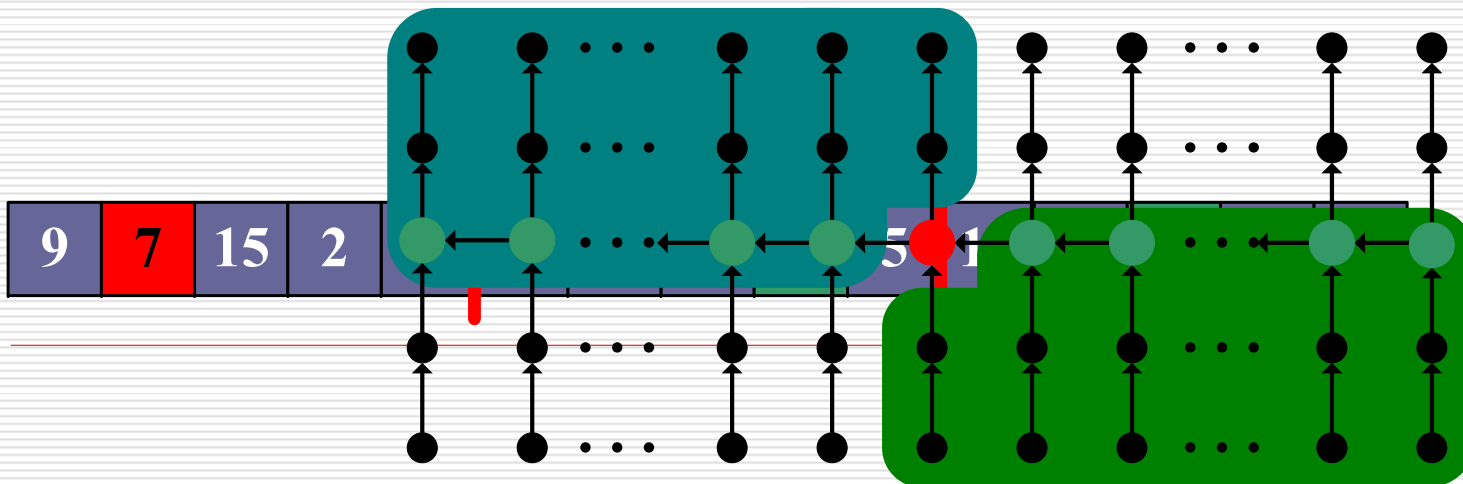


Ντετερμινιστική Επιλογή

- Χρόνος χειρότερης περίπτωσης:
 $T(n) \leq \Theta(n) + T(n/5) + T(7n/10), \quad T(1) = \Theta(1)$
- Λύση αναδρομής : $T(n) = \Theta(n)$
- Ντετερμινιστική επιλογή σε γραμμικό χρόνο!

Ενδιάμεσο Δείγματος

- Διαίρεση με ενδιάμεσο δείγματος σαν ρινοτ.
 - Μεγαλύτερος υποπίνακας $\leq 7n / 10$ στοιχεία.
 - Μικρότερος υποπίνακας $\geq 3n / 10$ στοιχεία.
- Ταξινομούμε 5αδες και βάζουμε σε αύξουσα σειρά των ενδιάμεσων στοιχείων τους (δείγματος).
- Ενδιάμεσος δείγματος στη $(n / 10)$ -οστή στήλη.
- Ενδιάμεσος δείγματος $\geq 3 \times n / 10$ στοιχεία.
 Ενδιάμεσος δείγματος $\leq 3 \times n / 10$ στοιχεία.



4	1	3
5	2	8
6	7	12
10	9	13
11	15	14

Σύνοψη

- Γρήγορη επιλογή (quickselect):
 - Πιθανοτικός αλγόριθμος με γραμμικό χρόνο (μ.π.)
 - Ντετερμινιστικός αλγόριθμος με γραμμικό χρόνο (χ.π.)
 - Ντετερμινιστικός αλγόριθμος με «bootstrapping»:
 - Για να βρω ενδιάμεσο για πολλά στοιχεία, βρίσκω «σχεδόν ενδιάμεσο» για λίγα.
 - Αυτό βοηθάει να βρω ενδιάμεσο για περισσότερα, ...

Ασκήσεις

- Τροποποίηση **quicksort** ώστε $O(n \log n)$ χρόνο σε χειρότερη περίπτωση. Είναι πρακτικό;
- Στον ντετερμινιστικό αλγόριθμο, χωρίζω **στοιχεία σε 3άδες (7άδες)**. Τι συμβαίνει;
- **A και B δύο ταξινομημένοι πίνακες** με n διαφορετικά στοιχεία ο καθένας. Σε χρόνο $O(\log n)$, το **ενδιάμεσο της ένωσης** των A και B.
- $T(n) = \Theta(n) + T(n/c) + T(n/d)$, $T(1) = \Theta(1)$
 - $1/c + 1/d < 1 \Rightarrow T(n) = \Theta(n)$
 - $1/c + 1/d = 1 \Rightarrow T(n) = \Theta(n \log n)$