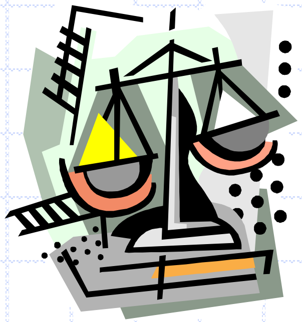
Επιλογή

Το πρόβλημα της επιλογής



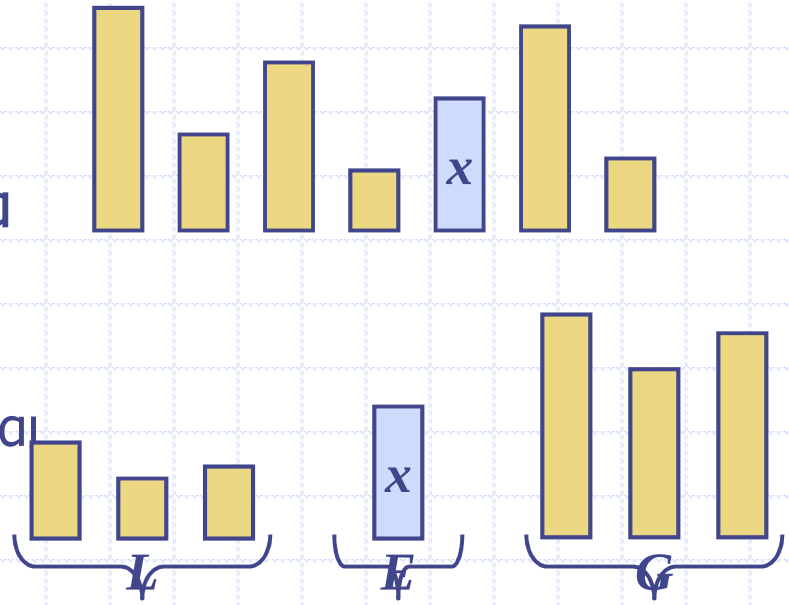
- Δοθέντος ενός ακεραίου k και n στοιχείων x_1, x_2, …, x_n, επιλεγμένων από μια ολική διάταξη, βρες το k τάξεως μικρότερο στοιχείο του συνόλου.
- Φυσικά, μπορούμε να ταξινομήσουμε τα στοιχεία σε O(n log n) χρόνο και να επιλέξουμε το στοιχείο με τον k δείκτη

k=3 $\boxed{7\ 4\ 9\ \underline{6}\ 2 \rightarrow 2\ 4\ \underline{6}\ 7\ 9}$

- Μπορούμε κάτι πιο γρήγορο?

Γρήγορη Επιλογή

- Η γρήγορη επιλογή είναι ένας τυχαίος αλγόριθμος επιλογής που βασίζεται στο παράδειγμα κλάδεμα και αναζήτηση:
  - Κλάδεμα: επιλογή ενός τυχαίου στοιχείου x (ονομάζεται pivot) και διαμέριση της S σε
    - ***L*: στοιχεία μικρότερα του *x***
    - ***E*: στοιχεία ίσα με *x***
    - ***G*: στοιχεία μεγαλύτερα του *x***
  - Αναζήτηση: ανάλογα με το k, η απάντηση είναι ή στο E, ή απαιτείται αναδρομή στο L ή το G



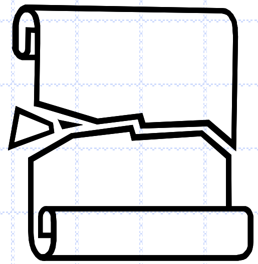
$k \le |L|$

$k > |L|+|E|$

$k' = k - |L| - |E|$

$|L| < k \le |L|+|E|$

(βρέθηκε)



Διαμέριση

- Διαμερίζουμε την ακολουθία όπως στην quick-sort:
  - Μεταφέρουμε, με την σειρά, κάθε στοιχείο y από την S και
  - Εισάγουμε το y στην L, E ή G, ανάλογα με το αποτέλεσμα της σύγκρισης με της σύγκρισης με το pivot x
- Κάθε εισαγωγή και διαγραφή γίνεται στην αρχή ή το τέλος της ακολουθίας και επομένως σε χρόνο $O(1)$
- Επομένως, το βήμα της διαμέρισης της quick-select απαιτεί χρόνο $O(n)$

```
Algorithm partition(S, p)
    Input sequence S, position p of pivot
    Output subsequences L, E, G of the
        elements of S less than, equal to,
        or greater than the pivot, resp.
    L, E, G ← empty sequences
    x ← S.remove(p)
    while ¬S.isEmpty()
        y ← S.remove(S.first())
        if y < x
            L.addLast(y)
        else if y = x
            E.addLast(y)
        else { y > x }
            G.addLast(y)
    return L, E, G
```

Οπτικοποίηση της Quick-Select

- Μια εκτέλεση της quick-select μπορεί να οπτικοποιηθεί από μια αναδρομική διαδρομή
  - Κάθε κόμβος αναπαριστά μια αναδρομική κλήση της quick-select, και αποθηκεύει το k και την υπόλοιπη ακολουθία

k=5, S=(7 4 9 3 2 6 5 1 8)

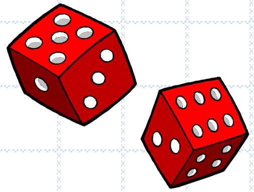
k=2, S=(7 4 9 6 5 8)

k=2, S=(7 4 6 5)

k=1, S=(7 6 5)

5

Μέσος Χρόνος τρεξίματος



- Έστω μια αναδρομική κλήση της quick-select σε μια ακολουθία μεγέθους s
  - Καλή κλήση: τα μεγέθη των L και G είναι το καθένα μικρότερο από $3s/4$
  - Κακή κλήση: μια από τις L και G έχει μέγεθος> από $3s/4$

7 2 9 4 3 7 <u>6</u> 1

2 4 3 1 | 7 9 7

Καλή κλήση

7 <u>2</u> 9 4 3 7 6 1

1 | 7 2 9 4 3 7 6

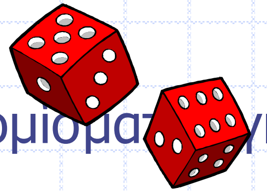
Κακή κλήση

- Μια κλήση είναι καλή με πιθανότητα $1/2$
  - 1/2 από πιθανά pivots δίνουν καλές κλήσεις:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

$\underbrace{1\ 2\ 3\ 4}_{\textbf{Κακά pivots}}\ \underbrace{5\ 6\ 7\ 8\ 9\ 10\ 11\ 12}_{\textbf{Καλά pivots}}\ \underbrace{13\ 14\ 15\ 16}_{\textbf{Κακά pivots}}$

Μέσος Χρόνος τρεξίματος 2



- Πιθανοτικό γεγονός #1: Το μέσο πλήθος στριψίματος νομίσματος για κορώνα είναι δυο
- Πιθανοτικό γεγονός #2: Η αναμενόμενη τιμή είναι γραμμική συνάρτηση:
  - $\boldsymbol{E(X + Y) = E(X) + E(Y)}$
  - $\boldsymbol{E(cX) = cE(X)}$
- Έστω T(n) ο αναμενόμενος χρόνος τρεξίματος της quick-select.
- Από το#2,
  - $T(n) \leq T(3n/4) + bn$*(αναμενόμενο # κλήσεων πριν από μια καλή κλήση)
- Από το t #1,
  - $T(n) \leq T(3n/4) + 2bn$
- Δηλαδή, T(n) είναι γεωμετρική σειρά:
  - $T(n) \leq 2bn + 2b(3/4)n + 2b(3/4)^2n + 2b(3/4)^3n + \ldots$
- Δηλ. η T(n) είναι O(n).
- Μπορούμε να επιλύσουμε το πρόβλημα της επιλογής σε μέσο χρόνο O(n).

Η γενική περίπτωση

$$T(n) \leq 2bn \sum_{i=0}^{\lceil \log_{4/3} n \rceil} (3/4)^i$$

Δηλαδή ο αναμενόμενος χρόνος είναι το πολύ 2bn επί ένα γεωμετρικό άθροισμα με βάση ένα θετικό μικρότερο του 1

Ντετερμινιστική επιλογή

- Μπορούμε να έχουμε επιλογή σε χρόνο O(n) στη χειρότερη.
- Βασική ιδέα: αναδρομική χρήση της επιλογής για να βρεθεί ένα καλό pivot της quick-select:
  - Διαμέριση της S σε n/5 σύνολα από 5 το καθένα
  - Εύρεση του ενδιάμεσου σε κάθε σύνολο
  - Αναδρομική εύρεση του ενδιάμεσου των παιδιών ενδιάμεσων.