

Αλγόριθμοι Αναζήτησης

Δημήτρης Φωτάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Πανεπιστήμιο Αιγαίου

Γραμμική Αναζήτηση

- Μοναδικός τρόπος όταν:
 - όχι ταξινομημένος πίνακας.
 - μόνο σειριακή προσπέλαση (π.χ. σειριακά αρχεία).
- ```
int linearSearch(int A[], int n, int x) {
 for (int i = 0; i < n; i++)
 if (x == A[i]) return(i);
 return(-1); }
```
- Χρόνος χ.π. / αποτυχημένης αναζήτησης:  $\Theta(n)$  (βέλτιστος).  
Χρόνος καλύτερης περίπτωσης:  $\Theta(1)$ .

Αναζήτηση 50:



Δομές Δεδομένων

Αλγόριθμοι Αναζήτησης 2

## Γραμμική Αναζήτηση (μ.π.)

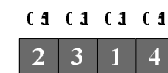
- Πιθανότητα αναζήτησης στοιχείου  $k = 1/n$ 
  - Όλα τα στοιχεία αναζητούνται ισοπίθانا!
- Χρόνος μ.π. =  $\sum_{i=k}^n P[k] \cdot k = \frac{n(n+1)}{2n} = \frac{n+1}{2}$
- Διαφορά από πιθανοτική quicksort:
  - RQS: η μέση περ. αναφέρεται στον πιθανοτικό αλγόριθμο.
  - LinearS: η μέση περ. αναφέρεται στην κατανομή των αναζητήσεων στα στοιχεία πίνακα!
- Δεν γνωρίζουμε κατανομή!

Δομές Δεδομένων

Αλγόριθμοι Αναζήτησης 3

## Γραμμική Αναζήτηση (μ.π.)

- Όχι ισοπίθانه αναζήτηση:
  - Μπροστά στοιχεία με μεγαλύτερη συχνότητα αναζήτησης.
- Δεν γνωρίζουμε συχνότητες:
  - Σταδιακή αναδιοργάνωση : μπροστά στοιχεία που ζητούνται.
  - Move-to-Front : χρόνος  $\leq 2 \times$  βέλτιστος, λίστα.
  - Move-Forward : πίνακας, όχι βελτίωση αν μόνο δύο στοιχεία.
- Self-organizing DS: «προσαρμόζεται» ώστε να είναι ταχύτερη στο μέλλον.



Μέσος # συγκρίσεων = 3.8

Δομές Δεδομένων

Αλγόριθμοι Αναζήτησης 4

## Διαδική Αναζήτηση

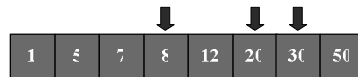
- Ταξινόμηση και τυχαία προσπέλαση.
- Χρόνος  $O(\log n)$  (βέλτιστος).

```

binarySearch(int A[], int n, int k) {
 int low = 0, up = n-1, mid;
 while (low <= up) {
 mid = (low + up) / 2;
 if (A[mid] == k) return (mid);
 else if (A[mid] > k) up = mid - 1;
 else low = mid + 1;
 }
 return (-1);
}

```

Αναζήτηση 30:



Δομές Δεδομένων

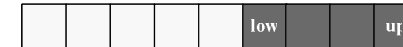
Αλγόριθμοι Αναζήτησης 5

## Ορθότητα

- Αν  $A[mid] > k$ ,  $k$  μπορεί να βρίσκεται μόνο αριστερά.
- Αν  $A[mid] < k$ ,  $k$  μπορεί να βρίσκεται μόνο δεξιά.
- Σε κάθε επανάληψη, πλήθος υποψήφιων στοιχείων μειώνεται (περίπου) στο μισό :

$$n, \frac{n}{2}, \frac{n}{4}, \dots, \frac{n}{2^{\ell}}, \dots, 1 \Rightarrow \ell \approx \log_2 n$$

- Χρόνος  $O(\log n)$ : βέλτιστος αν δεν γνωρίζουμε την κατανομή των στοιχείων.

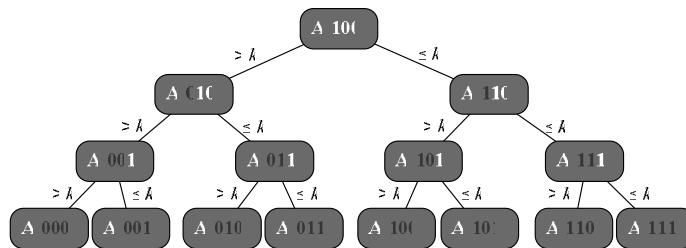


Δομές Δεδομένων

Αλγόριθμοι Αναζήτησης 6

## Χρόνος Εκτέλεσης

- Διαδική αναπαράσταση θέσεων: κάθε σύγκριση προσδιορίζει ένα bit της θέσης του στοιχείου!
- Διαδικό δέντρο συγκρίσεων έχει  $n$  φύλλα και  $\log n$  ύψος.

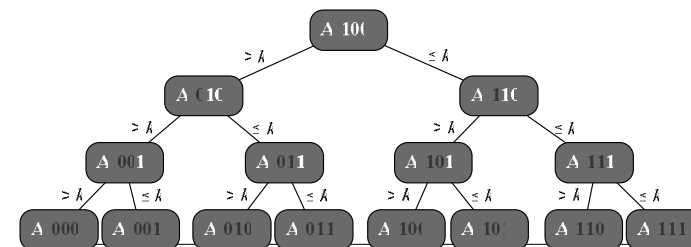


Δομές Δεδομένων

Αλγόριθμοι Αναζήτησης 7

## Χρόνος Εκτέλεσης

- Για κάθε αλγόριθμο, χωρίς άλλη πληροφορία (π.χ. κατανομή στοιχείων), μία σύγκριση προσδιορίζει ένα bit θέσης στοιχείου.
- Αν δεν γνωρίζουμε κατανομή στοιχείων, κάθε αλγόριθμος αναζήτησης χρειάζεται χρόνο  $\Omega(\log n)$  στη χειρότερη περίπτωση.



Δομές Δεδομένων

Αλγόριθμοι Αναζήτησης 8

## Αναζήτηση με Παρεμβολή

- Συνήθως έχουμε κάποια πληροφορία σχετικά με την κατανομή των στοιχείων (π.χ. όταν ψάχνουμε τον τηλεφωνικό κατάλογο δεν ανοίγουμε στη μέση).
- Παρεμβολή αξιοποιεί την πληροφορία κατανομής. Ταχύτερη (στη μέση περ.) από Δυαδική.
  - Επειδή ξέρουμε κατανομή, μια σύγκριση δίνει περισσότερο από ένα bit πληροφορίας σχετικά με θέση του στοιχείου.
  - Αναμενόμενη θέση  $k$  στο  $A[low \dots up]$ :
$$pos = low + (k - A[low]) / \text{μέση αύξηση τιμών ανά θέση}$$
  - Κατά τα άλλα ίδια με Δυαδική.

Δομές Δεδομένων

Αλγόριθμοι Αναζήτησης 9

## Αναζήτηση με Παρεμβολή

- Εκδοχή για ομοιόμορφη κατανομή σε διάστημα.
  - Προσαρμόζεται σε οποιαδήποτε άλλη κατανομή.

$$pos = low + (k - A[low]) \times \frac{up - low}{A[up] - A[low]}$$

Δομές Δεδομένων

Αλγόριθμοι Αναζήτησης 10

## Αναζήτηση με Παρεμβολή

- Χρόνος μέσης περίπτωσης :  $O(\log \log n)$ 
  - Αναζήτηση σε 1 τρισεκατ. στοιχεία με 6 συγκρίσεις!
- Χρόνος χειρότερης περίπτωσης :  $O(n)$
- Ο χρόνος χειρότερης περίπτωσης βελτιώνεται με Δυαδική Αναζήτηση με Παρεμβολή.
- Πρώτη επανάληψη βρίσκει περιοχή με ακρίβεια. Επόμενες προσπελάσεις στην ίδια περιοχή.
  - Καλή αξιοποίηση της cache memory.

Δομές Δεδομένων

Αλγόριθμοι Αναζήτησης 11

## Ασκήσεις

- Ακέραια τετραγωνική ρίζα φυσικού αριθμού  $n$  σε χρόνο  $O(\log n)$ .
  - Ακέραια τετραγωνική ρίζα  $n: x \in \mathbb{N} : x^2 \leq n \wedge (x+1)^2 > n$
- Ρίζα αύξουσας ακολουθίας στο  $\{1, \dots, n\}$  σε χρόνο  $O(\log n)$ .
- Δίνεται αύξουσα ακολουθία  $a_0, a_1, \dots, a_n$  (μπορεί να έχει και αρνητικούς όρους). Σε χρόνο  $O(\log n)$ , να βρεθεί αν υπάρχει ακέραιος  $i : a_i = i$

Δομές Δεδομένων

Αλγόριθμοι Αναζήτησης 12

## Ασκήσεις

---

- Μη ταξινομημένος πίνακας  $A$  με  $n$  διαφορετικούς αριθμούς και αριθμός  $x$ . Σε χρόνο  $O(n \log n)$ , να βρεθεί αν υπάρχει στον  $A$  ζεύγος στοιχείων με άθροισμα  $x$ .
- Ημίάθροισμα μέγιστου και ελάχιστου ανήκει στον πίνακα; Χρόνος εκτέλεσης (χ.π.) αν πίνακας  $(a)$  δεν είναι ταξινομημένος,  $(b)$  είναι ταξινομημένος, και  $(\gamma)$  έχει την ιδιότητα του σωρού;