ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

ΠΛΗΡΟΦΟΡΙΚΗ Ι

Ενότητα 8: Αναζήτηση και ταξινόμηση

Μιχάλης Δρακόπουλος

Σχολή Θετικών επιστημών

Τμήμα Μαθηματικών

Αναζήτηση (search)

Πρόβλημα: αναζήτηση της καταχώρησης `key` στη λίστα `LIST`
Μορφές αναζήτησης:

- Ύπαρξη συγκεκριμένης καταχώρησης στη λίστα.
- Θέση καταχώρησης στη λίστα.
- Συχνότητα εμφάνισης κάποιας καταχώρησης στη λίστα.

Η επιλογή αλγορίθμου αναζήτησης εξαρτάται από:

- Μέγεθος της λίστας.
- Διάταξη της λίστας.
- Πλήθος αναζητήσεων.

Η αποτελεσματικότητα της αναζήτησης καθορίζεται από τον αριθμό των συγκρίσεων που απαιτούνται σαν συνάρτηση του μεγέθους n της λίστας.

Αλγόριθμος γραμμικής αναζήτησης

```
function [pos, hit] = linear(key, LIST)
% hit - TRUE (1) αν υπάρχει key στη LIST
% pos - θέση του key στη LIST
n = length(LIST);          % πλήθος στοιχείων LIST
hit = false;
loc = 1;
while loc<=n && ~hit,
   if key == LIST(loc),
      pos = loc;
      hit = true;
   else
      loc = loc + 1;
   end
end
```

Γίνονται $O(n)$ συγκρίσεις κατά μέσο όρο.

Δυαδική binary αναζήτηση

Προυπόθεση η διάταξη της λίστας. Η σύγκριση του key με το μεσαίο στοιχείο της λίστας κατατάσσει την αναζητούμενη καταχώρηση σε ένα από τα δύο (αριστερό, δεξιό) τμήματα (υπολίστες) της LIST. Η διαδικασία επαναλαμβάνεται για το νέο τμήμα, που περιέχει τα μισά στοιχεία του προηγούμενου.
Ο μέγιστος αριθμός συγκρίσεων για τη δυαδική αναζήτηση εκφράζει ουσιαστικά
πόσες φορές το μέγεθος της λίστας n διαιρείται με το 2:

$$\lfloor \log_2 n \rfloor + 1$$

Αριθμός συγκρίσεων σε γραμμική και δυαδική αναζήτηση:

n	4	8	16	128	512	1024	1048576
γραμμική (μ.ο.)	2	4	8	64	256	512	524288
δυαδική (max)	3	4	5	8	10	11	21

Αλγόριθμος δυαδικής αναζήτησης

```
function [pos, hit] = binary(key, LIST)
n = length(LIST)
hit = false;
left = 1; right = n;
while ~hit && left<=right,
   mid = floor((left+right)/2);
   if key == LIST(mid),
      pos = mid;  hit = true;
   else
      if key < LIST(mid), right = mid - 1; else left = mid + 1; end
   end
end
```


Ταξινόμηση (sort)

- Διάταξη των στοιχείων μιας λίστας με βάση κάποια σχέση μεγέθους μεταξύ τους.
- Οι τεχνικές ταξινόμησης είναι ανεξάρτητες από το είδος διάταξης. Εδώ ζητείται η διάταξη κάποιας λίστας σε αύξουσα σειρά.
- Και η αποτελεσματικότητα της ταξινόμησης καθορίζεται από τον αριθμό των συγκρίσεων που απαιτούνται, σαν συνάρτηση του μεγέθους n της λίστας.

Ταξινόμηση με επιλογή (selection)

Έστω OLD η αρχική λίστα.

1. Δημιουργείται μια 2η λίστα NEW ίσου μεγέθους, αρχικά κενή.
2. Επιλέγεται το μικρότερο στοιχείο `small` της OLD.
3. Το `small` μεταφέρεται στην πρώτη διαθέσιμη θέση της NEW.
4. Η θέση του `small` στην OLD 'αδειάζει' (αντικαθίσταται από ένα πολύ μεγάλο αριθμό).
5. Επανάληψη των βημάτων 2, 3, και 4 έως ότου 'αδειάσει' η OLD.

Αλγόριθμος selection sort

Η ταξινόμηση με επιλογή πραγματοποιείται σε n βήματα (περάσματα) με $n-1$ συγκρίσεις σε κάθε πέρασμα.

```
function NEW = select(OLD)
n = length(OLD);
NEW = zeros(1,n);
for i = 1:n
   [small, pos] = min(OLD);
   NEW(i) = small;
   OLD(pos) = inf;
end
```


Αλγόριθμος selection sort (συνεχ.)

Χρειαζόμαστε επίσης συνάρτηση **min** που επιστρέφει το μικρότερο στοιχείο `small` μιας λίστας `LIST` καθώς και τη θέση του `pos` στη λίστα.

```
function [small, pos] = min(LIST)
n = length(LIST);
small = LIST(1);
pos = 1;
for i = 2:n
   if LIST(i) < small,
      small = LIST(i);
      pos = i;
   end
end
```

Ταξινόμηση με εναλλαγή (exchange)

Κατηγορία αλγορίθμων ταξινόμησης: συγκρίσεις ανά ζεύγη και εναλλαγή των τιμών τους για μερική διάταξη. Η ολική διάταξη επιτυγχάνεται μετά από μια σειρά κατάλληλων μερικών διατάξεων. Διαφοροποιούνται στη συστηματοποίηση του ελέγχου των ζευγών.
Ο αλγόριθμος της φυσαλίδας (bubble sort). Σε κάθε πέρασμα το max στοιχείο που απομένει στη λίστα 'αναδύεται' στη θέση $n - i + 1$.
Π.χ. το πρώτο πέρασμα για τη λίστα: 7, 9, 8, 3, 4 είναι:

βήμα 1ο: 7 9 8 3 4

βήμα 2ο: 7 9 8 3 4

βήμα 3ο: 7 8 9 3 4

βήμα 4ο: 7 8 3 9 4

βήμα 5ο: 7 8 3 4 | 9 |

Ο αλγόριθμος της φυσαλίδας (bubble sort) - I

```
function LIST = bubble(LIST)
n = max(size(LIST,1), size(LIST,2));
% maxz - μέγιστος αριθμός ζευγών σε κάθε πέρασμα
for maxz = n-1:-1:1,
   for z = 1:maxz
      if LIST(z) > LIST(z+1),
         temp = LIST(z);
         LIST(z) = LIST(z+1);
         LIST(z+1) = temp;
      end
   end
end
```


Πρόβλημα με τον αλγόριθμο I

... είναι δυνατό να έχει επιτευχθεί διάταξη προτού ολοκληρωθούν όλα τα περάσματα. STOP μετά από το πρώτο πέρασμα στο οποίο ΔΕΝ έγινε καμμία εναλλαγή...

- Χρησιμοποιούμε μια λογική μεταβλητή, έστω `xflag`.
- Αρχικά σε κάθε πέρασμα i η `xflag` είναι 0 (FALSE). Αν γίνει έστω και μια εναλλαγή στη διάρκεια του i τότε `xflag = 1`.
- Αν μετά το πέρασμα i, η τιμή της `xflag` παραμείνει 0, δεν έχουν γίνει εναλλαγές στοιχείων στο i και επομένως η λίστα είναι ήδη ταξινομημένη από το πέρασμα $i - 1$.

Ο αλγόριθμος της φυσαλίδας (bubble sort) - II

```
function LIST = bubble(LIST)
n = length(LIST);
maxz = n-1;
xflag = true; % =TRUE  υποθέτουμε ότι θα γίνουν εναλλαγές
while xflag
   xflag = false; % =FALSE  ΔΕΝ έχουν γίνει ακόμα εναλλαγές
   for z = 1:maxz
      if LIST(z) > LIST(z+1),
         temp = LIST(z);  LIST(z) = LIST(z+1);  LIST(z+1) = temp;
         xflag = true;
      end
   end
   maxz = maxz - 1;
end
```


Ανάλυση του αλγορίθμου ΙΙ

- Σε κάθε πέρασμα απαιτείται μια σύγκριση λιγότερη από το προηγούμενο πέρασμα.
- Χειρότερη περίπτωση όταν το ελάχιστο στοιχείο της λίστας βρίσκεται στην τελευταία θέση. Μέγιστος αριθμός συγκρίσεων:

$$(n-1)+(n-2)+\cdots+2+1=\frac{n(n-1)}{2}$$

- Όταν η λίστα είναι ήδη διατεταγμένη γίνονται $n-1$ συγκρίσεις.

Ταξινόμηση με εισαγωγή (insertion)

Έστω OLD η αρχική λίστα.

1. Δημιουργείται μια 2η λίστα NEW ίσου μεγέθους, αρχικά κενή.
2. Διατρέχονται τα στοιχεία της OLD με τη σειρά και εισάγονται κατάλληλα στη NEW, αφού ενδεχομένως μετακινηθούν κάποια από τα στοιχεία της NEW για να δημιουργήσουν θέση για την εισαγωγή του.

Ταξινόμηση με εισαγωγή (insertion) (συνεχ.)

Παράδειγμα:

Βήμα	OLD	NEW
1ο	$\underline{7}\ 9\ 8\ 3\ 4\ 7$	7 - - - - -
2ο	$7\ \underline{9}\ 8\ 3\ 4\ 7$	$7\ 9$ - - - -
3ο	$7\ 9\ \underline{8}\ 3\ 4\ 7$	$7\ 8\ 9$ - - -
4ο	$7\ 9\ 8\ \underline{3}\ 4\ 7$	$3\ 7\ 8\ 9$ - -
5ο	$7\ 9\ 8\ 3\ \underline{4}\ 7$	$3\ 4\ 7\ 8\ 9$ -
6ο	$7\ 9\ 8\ 3\ 4\ \underline{7}$	$3\ 4\ 7\ 7\ 8\ 9$

Προγραμματιστικά, η OLD διατρέχεται από αριστερά προς τα δεξιά (δείκτης i), ενώ η NEW από δεξιά προς αριστερά (δείκτης j).

Αλγόριθμος insertion sort

```
function NEW = insert(OLD)
n = length(OLD);
NEW = zeros(1,n);
for i = 1:n
   hit = false;
   j = i - 1;
   while j>0 & ∼hit,
      if NEW(j)>OLD(i),
         NEW(j+1) = NEW(j);
         j = j - 1;
      else
         hit = true;
      end
   end
   NEW(j+1) = OLD(i);
end
```


Ανάλυση αλγορίθμου εισαγωγής

- Σε κάθε βήμα i απαιτούνται κατά μ.ο. $(i-1)/2$ συγκρίσεις κατά τη γραμμική αναζήτηση στη NEW. Επομένως ο μ.ο. του αριθμού συγκρίσεων για όλα τα βήματα είναι:

$$(0+1+\cdots+(n-1))/2 = \frac{n(n-1)}{4}$$

- Χειρότερη περίπτωση όταν η λίστα είναι διατεταγμένη ανάποδα, με μέγιστο αριθμό συγκρίσεων:

$$\frac{n(n-1)}{2}$$

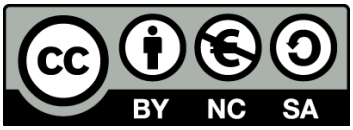
- Όταν η λίστα είναι ήδη διατεταγμένη απαιτούνται $n-1$ συγκρίσεις.

Σημειώματα

Σημείωμα Αναφοράς

Μιχάλης Δρακόπουλος. «Πληροφορική Ι. Ενότητα 8: Αναζήτηση και ταξινόμηση». Έκδοση: 1.0. Αθήνα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: http://opencourses.uoa.gr/modules/document/?course=MATH105.

Σημείωμα Αδειοδότησης



[1] http://creativecommons.org/licenses/by-nc-sa/4.0/

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

- Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:
- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.