Ελληνική Δημοκρατία
Τεχνολογικό Εκπαιδευτικό
Ίδρυμα Ηπείρου

Προγραμματισμός Ι

Ενότητα 8 : **Πίνακες ΙΙ**

Αλέξανδρος Τζάλλας

Τμήμα Μηχανικών Πληροφορικής Τ.Ε

Προγραμματισμός Ι

Ενότητα 8 : Πίνακες ΙΙ

Αλέξανδρος Τζάλλας
Λέκτορας

Άρτα, 2015

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- 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.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Ηπείρου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σκοποί ενότητας

- Να γίνουν κατανοητές οι έννοιες Ταξινόμηση & Αναζήτηση
- Να γίνει κατανοητή η χρησιμότητα χρήσης αλγορίθμων Ταξινόμησης & Αναζήτησης
- Να περιγραφεί ο αλγόριθμος ταξινόμησης bubblesort
- Να περιγραφεί ο αλγόριθμος ταξινόμησης selectionsort
- Να περιγραφεί ο αλγόριθμος Γραμμικής αναζήτηση (linear search)
- Να περιγραφεί ο αλγόριθμος Δυαδικής αναζήτηση (binary search)

Περιεχόμενα ενότητας

- Ταξινόμηση & Αναζήτηση
  - Ταξινόμηση bubblesort
  - Ταξινόμηση selectionsort
- Γραμμική αναζήτηση (linear search)
- Δυαδική αναζήτηση (binary search)

Ταξινόμηση & Αναζήτηση

- Δύο βασικές διαδικασίες επεξεργασίας πινάκων είναι η ταξινόμηση των στοιχείων ενός πίνακα και η αναζήτηση ενός συγκεκριμένου στοιχείου σε έναν πίνακα
- Στην καθημερινή ζωή συναντούμε πολλά προβλήματα ταξινόμησης η οποία γίνεται σύμφωνα με κάποιο κριτήριο (κατά αύξουσα τάξη, αλφαβητικά κ.α.)
- Τέτοια είναι η αλφαβητική ταξινόμηση των λέξεων ενός λεξικού, η ταξινόμηση των στοιχείων ενός πίνακα κατά αύξουσα σειρά, η ταξινόμηση των βιβλίων μιας βιβλιοθήκης και άλλα
- Για να ταξινομήσουμε τα στοιχεία ενός πίνακα υπάρχουν πολλοί αλγόριθμοι όπως η **bubblesort**, η mergesort. η quicksort, **selectionsort** και η insertinnsort

Ταξινόμηση bubblesort

- Για να κατανοήσουμε τη μέθοδο της φυσαλίδας (**bubblesort)** μπορούμε να φανταστούμε τα στοιχεία τον υπό ταξινόμηση **πίνακα σαν** φυσαλίδες η μία πάνω στην άλλη με τις τιμές του να **αντιστοιχούν στα βάρη** των αντίστοιχων φυσαλίδων
- Έτσι, οι ελαφρύτερες φυσαλίδες **θα έχουν την** τάση να ανέρχονται στην επιφάνεια ενώ οι βαρύτερες Θα έχουν την τάση να κατέρχονται προς το βυθό
- Η διαδικασία που ακολουθούμε είναι να τοποθετήσουμε το μικρότερο αριθμό στην πρώτη θέση, τον αμέσως μεγαλύτερο στη δεύτερη κ.ο.κ.
- Αυτό γίνεται μέσω διαδοχικών συγκρίσεων των στοιχείων τον πίνακα ανά δύο και ξεκινώντας από το τελευταίο με αποτέλεσμα τα μικρότερα στοιχεία να ανεβαίνουν προς τα πάνω σαν φυσαλίδες.

Παράδειγμα 1

- Να ταξινομηθούν τα στοιχεία τον παρακάτω πίνακα χρησιμοποιώντας τη μέθοδο της φυσαλίδας:

3	**6**	**21**	**16**	**57**

Να παρουσιαστούν οι ενδιάμεσες τιμές και το αποτέλεσμα)

Λύση$_{1/6}$

- Εφαρμόζοντας τη μέθοδο της φυσαλίδας ανασυντάσσουμε τα στοιχεία του δοσμένου πίνακα με 5 (= 6 - 1) περάσματα
- Ας δούμε τα περάσματα αυτά ένα-ένα:

1. Ξεκινάμε με το τελευταίο στοιχείο του πίνακα και διαδοχικά ελέγχουμε γειτονικά στοιχεία ανά δύο μέχρι να φτάσουμε στα πρώτα δύο στοιχεία
2. Μόλις εντοπίσουμε ένα ζεύγος στοιχείων στο οποίο η κάτω τιμή είναι μικρότερη από την πάνω αντιμεταθέτουμε τις τιμές τον πίνακα ώστε η μικρότερη τιμή να ανέβει προς τα πάνω
3. Για κάθε πέρασμα γίνονται 5 (=6 -1) έλεγχοι για πιθανές αντιμεταθέσεις
4. Στο πρώτο πέρασμα η μικρότερη τιμή θα είναι στην πρώτη θέση, στο δεύτερο πέρασμα η δεύτερη πιο μικρή τιμή θα είναι στη δεύτερη θέση κ.ο.κ.
5. Τα πέντε περάσματα είναι τα εξής:

Λύση$_{2/6}$
Πρώτο Πέρασμα

4	4	4	4	**4**	2
7	7	7	**7**	**2**	4
12	12	**12**	**2**	7	7
2	**2**	**2**	12	12	12
5	**3**	3	3	3	3
3	5	5	5	5	5

Λύση$_{3/6}$
Δεύτερο Πέρασμα

2	2	2	2	2
4	4	4	**4**	3
7	7	**7**	**3**	4
12	**12**	**3**	7	7
3	**3**	12	12	12
5	5	5	5	5

Λύση$_{4/6}$
Τρίτο Πέρασμα

2	2	2	2
3	3	3	3
4	4	**4**	4
7	**7**	**5**	5
12	**5**	7	7
5	12	12	12

Λύση$_{5/6}$
Τέταρτο Πέρασμα

2	2	2
3	3	3
4	4	4
5	**5**	5
7	**7**	7
12	12	12

Λύση$_{6/6}$

Πέμπτο Πέρασμα

2	2
3	3
4	4
5	5
7	7
12	12

Άρα ο ταξινομημένος πίνακας είναι: **2 3 4 5 7 12**

Παράδειγμα 2

- Να γραφεί πρόγραμμα, για την ταξινόμηση σε αύξουσα σειρά των στοιχείων ενός πίνακα Ν στοιχείων. Να γίνει εφαρμογή για έναν πίνακα ακέραιων αριθμών.

Λύση

```
program anazitisi2(input,output);
var a: array [ 1..50] of real;
x: real;
i: integer;
k: boolean;
begin
writeln ('dwse to stoixeio pou thes na exetaseis ' );
readln(x);
writeln ('dwse 50 stoixeia:  ');
for i:= 1 to 50 do
begin
write('a[',i,']=');
readln(a[i]);
end;
k:=false;
for i:=1 to 50 do
if a[i]=x then
begin
k:=true;
writeln(' To stoixeio brethike sth thesi: ', i);
end;
if k=false then
writeln(' To stoixeio de brethike ')
end.
```

Ταξινόμηση selectionsort 1/5

- Ας υποθέσουμε ότι έχουμε τον εξής πίνακα τον οποίο θέλουμε να ταξινομήσουμε με τη μέθοδο της **selectionsort**:

50	9	28	42	15	62	30	5

- **i=1:** Στην πρώτη επανάληψη εκτέλεσης αυτό που πρέπει να κάνουμε είναι να θεωρήσουμε ότι η ροή τον ελέγχου βρίσκεται στο πρώτο στοιχείο ενώ το όριο (η περιοχή) στο οποίο θα αναζητήσουμε το ελάχιστο είναι ολόκληρος ο πίνακας
- Εδώ το ελάχιστο στοιχείο του πίνακα είναι το 5 το οποίο το εναλλάσσουμε με το πρώτο στοιχείο τον πίνακα στο οποίο βρίσκεται ο έλεγχος
- Έτσι προκύπτει ένας νέος πίνακας ο οποίος στη θέση του 50 έχει το 5 και αντίστροφα, ενώ όλα τα υπόλοιπα στοιχεία είναι τα ίδια. Ο νέος πίνακας είναι ο εξής:

5	9	28	42	15	62	30	50

Ταξινόμηση selectionsort 2/5

- **i=2:** Στη δεύτερη επανάληψη θεωρούμε ότι η ροή του ελέγχου βρίσκεται στο δεύτερο στοιχείο ενώ το όριο στο οποίο θα αναζητήσουμε το ελάχιστο είναι ολόκληρος ο πίνακας εκτός από το πρώτο στοιχείο που είναι πλέον ήδη ταξινομημένο.
- Το ελάχιστο στοιχείο του υπόλοιπου πίνακα είναι το 9 το οποίο δεν εναλλάσσουμε με κάποιο άλλο στοιχείο αφού είναι ήδη το ελάχιστο τον αντίστοιχου υποπίνακα (δηλαδή όλον τον πίνακα εκτός από το 5). Έτσι ο πίνακας παραμένει ο ίδιος:

5	9	28	42	15	62	30	50

Ταξινόμηση selectionsort$_{3/5}$

- **i=3**: Στην τρίτη επανάληψη θεωρούμε ότι η ροή ελέγχου βρίσκεται στο τρίτο στοιχείο ενώ το όριο στο οποίο Θα αναζητήσουμε το ελάχιστο είναι ολόκληρος ο πίνακας εκτός από τα δύο πρώτα στοιχεία που είναι ήδη ταξινομημένα
- Το ελάχιστο στοιχείο του υπόλοιπου πίνακα είναι το 15 το οποίο εναλλάσσουμε με το 28. ‘Ετσι ο νέος πίνακας είναι ο εξής:

5	9	15	28	42	62	30	50

Ταξινόμηση selectionsort 4/5

- **i=4**: Ομοίως με πριν αντιμεταθέτουμε το 42 με το 28 και έχουμε:

5	**9**	**15**	**28**	42	62	30	50

- **i=5**: Ομοίως με πριν αντιμεταθέτουμε το 42 με το 30 και έχουμε:

5	**9**	**15**	**28**	**30**	62	42	50

- **i=6**: Ομοίως με πριν αντιμεταθέτουμε το 62 με το 42 και έχουμε:

5	**9**	**15**	**28**	**30**	**42**	62	50

Ταξινόμηση selectionsort$_{5/5}$

- **i=7**: Ομοίως με πριν αντιμεταθέτουμε το 62 με το 50 και έχουμε:

5	9	15	28	30	42	50	**62**

- Επειδή σε αυτό το σημείο ο πίνακας θα είναι πάντοτε ταξινομημένος, δε χρειάζεται να ελέγξουμε και το τελευταίο στοιχείο (π.χ. σε αυτό το σημείο το 62 βρίσκεται ήδη στη σωστή του θέση). Επομένως ο τελικός πίνακας είναι ο εξής:

5	9	15	28	30	42	50	62

Αναζήτηση στοιχείου σε πίνακα

- 'Ενα πολύ συνηθισμένο πρόβλημα κατά τη χρήση πινάκων είναι να διαπιστώσουμε αν υπάρχει μία συγκεκριμένη τιμή σε κάποιον πίνακα, και εφόσον υπάρχει, τη θέση της στον πίνακα
- Για την αντιμετώπιση του προβλήματος αυτού μπορούμε να χρησιμοποιήσουμε κάποιον από τους αλγορίθμους αναζήτησης
- Οι κυριότεροι από αυτούς είναι η γραμμική (σειριακή) αναζήτηση και η δυαδική αναζήτηση

Γραμμική αναζήτηση (linear search)$_{1/3}$

- Υπάρχουν τρεις βασικές περιπτώσεις για τη γραμμική αναζήτηση:
- **Γραμμική αναζήτηση σε πίνακα μη ταξινομημένο στον οποίο κάθε στοιχείο να είναι διαφορετικό από όλα τα υπόλοιπα**
  - Σε αυτή την περίπτωση εξετάζουμε ένα-ένα όλα τα στοιχεία τον πίνακα ξεκινώντας από την αρχή μέχρι να βρεθεί το ζητούμενο στοιχείο
  - Για να σταματάει ο έλεγχος όταν βρεθεί η ζητούμενη τιμή θεωρούμε μία λογική μεταβλητή η οποία τότε γίνεται true
  - Στην αρχή αυτή η μεταβλητή υποθέτουμε ότι έχει τιμή false και αν η τιμή της παραμείνει false μέχρι το τέλος του ελέγχου τότε αυτό σημαίνει ότι δεν υπάρχει αυτή η τιμή στον πίνακα, αλλιώς αν είναι true τότε η τιμή έχει βρεθεί

Παράδειγμα 3

- Αν υποθέσουμε ότι έχουμε ένα πίνακα 50 πραγματικών στοιχείων, τότε το πρόγραμμα με το οποίο αναζητούμε ένα στοιχείο στον πίνακα το οποίο το δίνει ο χρήστης είναι το εξής:

Λύση

```
program anazitisi1(input,output);
var a: array [ 1..50] of real;
x: real;
i,j: integer;
k: boolean;
begin
writeln ('dwse to stoixeio pou thes na anazhthseis ' );
readln(x);
writeln ('dwse 50 stoixeia: ');
for i:= 1 to 50 do
readln(a[i]);
k:=false;j:=0;i:=1;
while (i <= 50) and (k = false) do
begin
if a[i] =x then
begin
j:= i;
k:= true;
end;
i:= i+1;
end;
if k = true then
writeln ('to stoixeio brethike ston pinaka sth thesi: ', j)
else
writeln ('to stoixeio de brethike ston pinaka')
end.
```

Γραμμική αναζήτηση (linear search)$_{2/3}$

- **Γραμμική αναζήτηση σε πίνακα μη ταξινομημένο για τον οποίο δε γνωρίζουμε αν κάθε στοιχείο τον υπάρχει και άλλες φορές στον πίνακα**
- Με αυτή τη μέθοδο εξετάζονται όλα τα στοιχεία τον πίνακα από το πρώτο μέχρι το τελευταίο σε αντίθεση με την προηγούμενη περίπτωση που ο έλεγχος τελείωνε μόλις βρίσκαμε το ζητούμενο στοιχείο
- Ομοίως με πριν και εδώ κατά την αναζήτηση του ζητούμενου στοιχείου μόλις βρεθεί κάποιο στοιχείο που έχει τη ζητούμενη τιμή το πρόγραμμα εμφανίζει τη Θέση τον στην οθόνη
- Ο έλεγχος όμως συνεχίζεται για την περίπτωση που υπάρχει η ίδια τιμή και σε άλλες θέσεις τον πίνακα οπότε αυτές εμφανίζονται. Το πρόγραμμα για την εύρεση ενός αριθμού που εισάγει ο χρήστης από έναν πίνακα πενήντα πραγματικών αριθμών είναι το εξής…

Παράδειγμα 4

- Το πρόγραμμα για την εύρεση ενός αριθμού που εισάγει ο χρήστης από έναν πίνακα πενήντα πραγματικών αριθμών είναι το εξής:

Λύση

```
program anazitisi1(input,output);
var a: array [ 1..50] of real;
x: real;
i,j: integer;
k: boolean;
begin
writeln ('dwse to stoixeio pou thes na anazhthseis ' );
readln(x);
writeln ('dwse 50 stoixeia: ');
for i:= 1 to 50 do
readln(a[i]);
k:=false;j:=0;i:=1;
while (i <= 50) and (k = false) do
begin
if a[i] =x then
begin
j:= i;
k:= true;
end;
i:= i+1;
end;
if k = true then
writeln ('to stoixeio brethike ston pinaka sth thesi: ', j)
else
writeln ('to stoixeio de brethike ston pinaka')
end.
```

Γραμμική αναζήτηση (linear search)$_{3/3}$

- **Γραμμική αναζήτηση σε πίνακα με τα στοιχεία τον ταξινομημένα σε αύξουσα σειρά**
  - Σε αυτήν την περίπτωση ξεκινώντας από το στοιχείο στην πρώτη θέση ελέγχουμε αν κάποιο από τα στοιχεία τον πίνακα συμπίπτει με το στοιχείο που αναζητάμε
  - Όμως εδώ ο έλεγχος σταματά μόλις βρεθεί στοιχείο με τιμή μεγαλύτερη από την τιμή αναζήτησης.
  - Αν η τιμή ενός στοιχείου κατά τη διάρκεια της αναζήτησης είναι ίση με την τιμή αναζήτησης τότε στην οθόνη εμφανίζεται η θέση τον και εξετάζεται το επόμενο στοιχείο τον πίνακα
  - Αν αυτό είναι μεγαλύτερο τότε η αναζήτηση τελειώνει.
  - Αν η ζητούμενη τιμή δεν υπάρχει στον πίνακα, τότε πάλι η αναζήτηση θα τελειώσει μόλις βρεθεί το αμέσως μεγαλύτερο στοιχείο τον πίνακα από το ζητούμενο και στην οθόνη θα εμφανιστεί αντίστοιχο μήνυμα.

Παράδειγμα 5

- Το πρόγραμμα για την αναζήτηση ενός πραγματικού αριθμού από έναν πίνακα πενήντα στοιχείων είναι το εξής:

Λύση

```
program anazitisi3(input,output);
var a: array [ 1..50] of real;
x: real;
i,j: integer;
k: boolean;
begin
writeln ('dwse to stoixeio pou thes na exetaseis ' );
readln(x);
writeln ('dwse 50 stoixeia:  ');
for i:= 1 to 50 do
begin
write('a[',i,']=');
readln(a[i]);
end;
k:=false;j:=0;i:=1;
while (i<=100) and (x<=a[i]) do
begin
if a[i]=x then
begin
j:=i;
k:=true;
writeln(' H thesi toy stoixeioy ston pinaka einai: ', j);
end;
i:=i+1;
end;
if k=false then
writeln(' To stoixeio de brethike ')
end.
```

Δυαδική αναζήτηση (binary search)$_{1/2}$

- Η δυαδική αναζήτηση είναι μία μέθοδος αναζήτησης ενός στοιχείου σε ένα πίνακα, η οποία εφαρμόζεται μόνο στην περίπτωση που τα στοιχεία ενός πίνακα είναι ταξινομημένα σε αύξουσα σειρά
- Σύμφωνα με τη μέθοδο δυαδικής αναζήτησης αποκλείουμε διαδοχικά ολόκληρα τμήματα του πίνακα στα οποία δε γίνεται κάθε φορά να βρίσκεται το ζητούμενο στοιχείο
- Σε αντίθεση με τη γραμμική αναζήτηση εδώ αρχικά εξετάζουμε το μεσαίο στοιχείο του πίνακα.

Δυαδική αναζήτηση (binary search)$_{2/2}$

- Επειδή ο πίνακας είναι ταξινομημένος κατ' αύξουσα σειρά, αν το ζητούμενο στοιχείο είναι μεγαλύτερο από το μεσαίο, μπορούμε μπορούμε με βεβαιότητα να αποκλείσουμε την περίπτωση να βρίσκεται στο πρώτο μισό τμήμα του πίνακα, ενώ αν είναι μικρότερο από το μεσαίο μπορούμε ομοίως να αποκλείσουμε την περίπτωση να βρίσκεται στο δεύτερο μισό του πίνακα
- Συγκεκριμένα, εδώ εξετάζουμε το μεσαίο στοιχείο αυτού του τμήματος και ελέγχουμε ομοίως με πριν αν το ζητούμενο στοιχείο είναι μικρότερο ή μεγαλύτερο από το μεσαίο περιορίζοντας πάλι το τμήμα του πίνακα που ελέγχουμε στο μισό
- Αυτή η διαδικασία συνεχίζεται μέχρι να εντοπίσουμε το ζητούμενο στοιχείο ή μέχρι να αποκλείσουμε ολόκληρο τον πίνακα

Βιβλιογραφία

Βλαχάβας Ι. (1994). Η γλώσσα προγραμματισμού Pascal. Εκδόσεις Γαρταγάνης Διονύσιος.

Κάβουρας Ι.Κ. (1999). Δομημένος Προγραμματισμός με Pascal. Εκδόσεις Κλειδάριθμος.

Αλεβίζου Θ., & Καμπουρέλης Α. (1995). Μαθήματα Προγραμματισμού: Εισαγωγή με τη Γλώσσα Pascal. Εκδόσεις Παπασωτηρίου.

Cooper D. (1993). Oh! Pascal!, An Introduction to Computing, του. Εκδόσεις Norton.

Larry R.N. (1998). Advanced Programming in Pascal with Data Structures. Εκδόσεις Macmillan USA.

Τσελίκης Γ.Σ., Τσελίκας Ν.Δ. (2012). C: από τη Θεωρία στην Εφαρμογή (Β' Έκδοση). Εκδόσεις Παπασωτηρίου.

Aho A.V., Hopcroft J.E., & Ullman J.D. (1974). The design and analysis of computer algorithms. Εκδόσεις Addison Wesley.

Abelson H., Sussman G.J., Sussman J. (1985). Structure and Interpretation of Computer Programs, MIT Press, McGraw Hill Book Company.

Σημείωμα Αναφοράς

Σημείωμα Αδειοδότησης

[1] http://creativecommons.org/licenses/by-nc-nd/4.0/deed.el

Τέλος Ενότητας

Επεξεργασία: Ευάγγελος Καρβούνης

Άρτα, 2015

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επενδύοντας στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Τέλος Ενότητας

Πίνακες ΙΙ