



ΠΛΗΡΟΦΟΡΙΚΗ & ΑΛΓΟΡΙΘΜΟΙ

Ενότητα # (6): Αλγοριθμική – Δομές Δεδομένων

Κύδρος Δημήτρης
Τμήμα Λογιστικής και Χρηματοοικονομικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



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 παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Κεντρικής Μακεδονίας» 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.



Αλγοριθμική – Θεμελιώδεις Δομές δεδομένων

Περιεχόμενα ενότητας

1. Γιατί Δομές Δεδομένων?
2. Βασικές δομές δεδομένων
3. Στατικές δομές δεδομένων
4. Εφαρμογές

Σκοποί ενότητας

- Να γνωρίσετε τις βασικές στατικές δομές δεδομένων
- Να χρησιμοποιήσετε τις δομές δεδομένων σε συγκεκριμένα προβλήματα

Γιατί δομές δεδομένων?

ΑΛΓΟΡΙΘΜΟΣ Δοκιμή

ΜΕΤΑΒΛΗΤΕΣ

x, i, plithos, S: **ΑΚΕΡΑΙΕΣ**

ΜΟ: **ΠΡΑΓΜΑΤΙΚΕΣ**

ΑΡΧΗ

ΕΠΑΝΑΛΑΒΕ

ΔΙΑΒΑΣΕ plithos

ΜΕΧΡΙ plithos>0

S←0

ΓΙΑ i←1 **ΜΕΧΡΙ** plithos **ΑΡΧΗ**

ΔΙΑΒΑΣΕ x

S←S+x

ΤΕΛΟΣ

ΜΟ←S/plithos

//σε αυτό το σημείο ο μέσος όρος έχει υπολογιστεί

ΤΕΛΟΣ

«Να γίνει αλγόριθμος που θα διαβάσει ένα γνωστό πλήθος από ακέραιους αριθμούς και θα βρίσκει το μέσο όρο τους. Στη συνέχεια να βρεθεί η απόσταση (σε ακέραιες τιμές) του κάθε αριθμού από τον υπολογισμένο μέσο όρο»

για τη λύση αυτού του, απλού φαινομενικά, προβλήματος απαιτείται ένας τρόπος αποθήκευσης πολλών ομοειδών τιμών σε μία ενιαία κατασκευή. Αυτές οι κατασκευές ονομάζονται **δομές δεδομένων**.

ΟΡΙΣΜΟΙ - ΠΡΑΞΕΙΣ

Μία δομή δεδομένων είναι μια οργανωμένη συλλογή από ομοειδή αποθηκευμένα δεδομένα. Στα δεδομένα αυτά μπορούν να εφαρμοστούν διάφορες λειτουργίες οι οποίες θα τα επεξεργάζονται.

Προσπέλαση, Εισαγωγή, Διαγραφή, Αναζήτηση, Ταξινόμηση, Αντιγραφή, Συγχώνευση, Διαχωρισμός, Τεμαχισμός κλπ

Στατικές - Δυναμικές

Στατικές:

1. Συγκεκριμένο πλήθος
2. Συνεχόμενες Θέσεις
Μνήμης

Δυναμικές:

1. Όχι Συγκεκριμένο πλήθος
2. Μη Συνεχόμενες Θέσεις
Μνήμης

ΠΙΝΑΚΕΣ (στατική δομή)

<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
'α'	'ζ'	'Ν'	'ο'	'σ'	'Κ'

Πίνακας με όνομα Α

	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
<i>1</i>	2	4	1	-1
<i>2</i>	8	3	4	2
<i>3</i>	1	1	0	2

Πίνακας με όνομα Β

Μονοδιάστατος,
δισδιάστατος, τρισδιάστατος
κλπ

Προσπέλαση

<p>ΑΛΓΟΡΙΘΜΟΣ Εισ_Εξ ΜΕΤΑΒΛΗΤΕΣ <i>i</i>, A[10]: ΑΚΕΡΑΙΕΣ ΑΡΧΗ ΓΙΑ <i>i</i>←1 ΜΕΧΡΙ 10 ΑΡΧΗ ΔΙΑΒΑΣΕ A[<i>i</i>] ΤΕΛΟΣ //..... ΓΙΑ <i>i</i>←1 ΜΕΧΡΙ 10 ΑΡΧΗ ΓΡΑΨΕ A[<i>i</i>] ΤΕΛΟΣ ΤΕΛΟΣ</p>	<pre>#include <stdio.h> int main(void){ int i, A[11]; //οι πίνακες ξεκινούν από το 0 στη C for(i=1;i<=10;i++){ scanf("%d", &A[i]); fflush(stdin); } //..... for(i=1;i<=10;i++){ printf("%d ", A[i]); } }</pre>
---	--

Και η επίλυση του αρχικού προβλήματος

```
ΑΛΓΟΡΙΘΜΟΣ Δοκιμή_1
ΜΕΤΑΒΛΗΤΕΣ
    Χ[10], i, S: ΑΚΕΡΑΙΕΣ
    ΜΟ: ΠΡΑΓΜΑΤΙΚΕΣ
ΑΡΧΗ
    S ← 0
    ΓΙΑ i ← 1 ΜΕΧΡΙ 10 ΑΡΧΗ
        ΔΙΑΒΑΣΕ x[i]
        S ← S + x[i]
    ΤΕΛΟΣ
    ΜΟ ← S / 10
    ΓΙΑ i ← 1 ΜΕΧΡΙ 10 ΑΡΧΗ
        ΓΡΑΨΕ ΜΟ - x[i]
    ΤΕΛΟΣ
ΤΕΛΟΣ
```

Δισδιάστατοι πίνακες

	1	2	3	4
1	2	4	1	-1
2	8	3	9	2
3	1	1	0	2

<p>ΑΛΓΟΡΙΘΜΟΣ Εισ_Εξ_κατά_Γραμμή ΜΕΤΑΒΛΗΤΕΣ i, j, A[10,5]: ΑΚΕΡΑΙΕΣ</p> <p>ΑΡΧΗ ΓΙΑ i←1 ΜΕΧΡΙ 10 ΑΡΧΗ ΓΙΑ j←1 ΜΕΧΡΙ 5 ΑΡΧΗ ΔΙΑΒΑΣΕ A[i, j] ΤΕΛΟΣ ΤΕΛΟΣ //..... ΓΙΑ i←1 ΜΕΧΡΙ 10 ΑΡΧΗ ΓΙΑ j←1 ΜΕΧΡΙ 5 ΑΡΧΗ ΓΡΑΨΕ A[i, j] ΤΕΛΟΣ ΤΕΛΟΣ ΤΕΛΟΣ</p>	<pre>#include <stdio.h> int main(void){ int i, j, A[11][6]; //οι πίνακες ξεκινούν από το 0 στη C for(i=1;i<=10;i++){ for(j=1;j<=5;j++){ scanf("%d", &A[i][j]); fflush(stdin); } } //..... for(i=1;i<=10;i++){ for(j=1;j<=5;j++){ printf("%d ", A[i][j]); } } }getchar(); }</pre>
<p>ΑΛΓΟΡΙΘΜΟΣ Εισ_Εξ_κατά_Στήλη ΜΕΤΑΒΛΗΤΕΣ i, j, A[10,5]: ΑΚΕΡΑΙΕΣ</p> <p>ΑΡΧΗ ΓΙΑ j←1 ΜΕΧΡΙ 5 ΑΡΧΗ ΓΙΑ i←1 ΜΕΧΡΙ 10 ΑΡΧΗ ΔΙΑΒΑΣΕ A[i, j] ΤΕΛΟΣ ΤΕΛΟΣ //..... ΓΙΑ j←1 ΜΕΧΡΙ 5 ΑΡΧΗ ΓΙΑ i←1 ΜΕΧΡΙ 10 ΑΡΧΗ ΓΡΑΨΕ A[i, j] ΤΕΛΟΣ ΤΕΛΟΣ ΤΕΛΟΣ</p>	<pre>#include <stdio.h> int main(void){ int i, j, A[11][6]; //οι πίνακες ξεκινούν από το 0 στη C for(j=1;j<=5;j++){ for(i=1;i<=10;i++){ scanf("%d", &A[i][j]); fflush(stdin); } } //..... for(j=1;j<=5;j++){ for(i=1;i<=10;i++){ printf("%d ", A[i][j]); } } }getchar(); }</pre>

Ειδικές περιπτώσεις

1. τετραγωνικός. Για παράδειγμα, ένας πίνακας $A_{k \times k}$ είναι τετραγωνικός καθώς έχει k γραμμές και k στήλες (k γνωστό).

Σε έναν τετραγωνικό πίνακα, τα στοιχεία που έχουν τον ίδιο δείκτη γραμμής και στήλης, είναι δηλαδή της μορφής $A[i, i]$, $i=1, 2, \dots, k$, ανήκουν στην **κύρια διαγώνιο**. Έτσι, στον πίνακα $A_{k \times k}$ τα στοιχεία $A[1, 1]$, $A[2, 2]$, $A[3, 3]$, κλπ ανήκουν στην κύρια διαγώνιο του πίνακα.

Σε έναν τετραγωνικό πίνακα $A_{k \times k}$, τα στοιχεία $A[i, k-i+1]$, ανήκουν στη **δευτερεύουσα διαγώνιο** του πίνακα.

Σε έναν τετραγωνικό πίνακα $A_{k \times k}$, όταν $A[i, j]=A[j, i]$, για κάθε $i, j=1, 2, \dots, k$, τότε ο πίνακας είναι **συμμετρικός** ως προς την κύρια διαγώνιο του.

2. Ένας τετραγωνικός πίνακας που περιέχει **μηδενικά** (0) σε όλα τα στοιχεία $A[i, j]$, για $i > j$, λέγεται **άνω τριγωνικός**.

3. Αντίστοιχα, ένας **κάτω τριγωνικός** πίνακας περιέχει μηδενικά (0) σε όλα τα στοιχεία $A[i, j]$, $i < j$.

4. Ένας τετραγωνικός πίνακας $A_{k \times k}$ λέγεται **κεντροσυμμετρικός**, όταν όλα τα στοιχεία του είναι συμμετρικά ίσα ως προς το κέντρο του πίνακα. Στους κεντροσυμμετρικούς πίνακες ισχύει: $A[i, j] = A[k-i+1, k-j+1]$.

Σχηματικά.....

3	2	4	5
1	3	9	1
1	2	0	2
7	4	3	2

Τετραγωνικός πίνακας

3	2	4	5
1	3	9	1
1	2	0	2
7	4	3	2

Στοιχεία της κυρίας διαγωνίου

3	2	4	5
1	3	9	1
1	2	0	2
7	4	3	2

Στοιχεία της δευτερεύουσας διαγωνίου

3	2	4	5
2	3	9	1
4	9	0	3
5	1	3	2

Συμμετρικός πίνακας

3	2	4	5
0	3	9	1
0	0	1	3
0	0	0	2

Άνω τριγωνικός πίνακας

3	0	0	0
2	3	0	0
4	9	1	0
5	1	3	2

Κάτω τριγωνικός πίνακας

2	3	4	6
5	7	0	9
9	0	7	5
6	4	8	2

Κεντροσυμμετρικός πίνακας με άρτιο
πλήθος γραμμών και στηλών
(το κέντρο είναι στην τομή των διπλών
γραμμών)

2	8	4	6	5
7	0	9	10	1
3	5	12	5	3
1	10	9	0	7
5	6	4	8	2

Κεντροσυμμετρικός πίνακας με περιττό
πλήθος γραμμών και στηλών
(το κέντρο είναι το στοιχείο [3,3])

Άθροισμα

ΑΛΓΟΡΙΘΜΟΣ ΥΠΟΛΟΓΙΣΜΟΙ

ΜΕΤΑΒΛΗΤΕΣ

X[10], Σ_Θ, Σ_A, ΠΛΗΘΟΣ_A, i: **ΑΚΕΡΑΙΕΣ**

M_O_A: **ΠΡΑΓΜΑΤΙΚΕΣ**

ΑΡΧΗ

Σ_Θ ← 0

//αρχικοποίηση αθροίσματος θετικών

Σ_A ← 0

//αρχικοποίηση αθροίσματος αρνητικών

ΠΛΗΘΟΣ_A ← 0

//αρχικοποίηση πλήθους αρνητικών

ΓΙΑ i ← 1 **ΜΕΧΡΙ** 10 **ΑΡΧΗ**

ΔΙΑΒΑΣΕ X[i]

ΑΝ X[i] > 0 **ΤΟΤΕ**

Σ_Θ ← Σ_Θ + X

ΑΛΛΙΩΣ

ΑΝ X[i] < 0 **ΤΟΤΕ ΑΡΧΗ**

Σ_A ← Σ_A + X[i]

ΠΛΗΘΟΣ_A ← ΠΛΗΘΟΣ_A + 1

ΤΕΛΟΣ

ΤΕΛΟΣ

//επανάληψης και εσωτερικών υπολογισμών

ΓΡΑΨΕ Σ_Θ

ΑΝ ΠΛΗΘΟΣ_A <> 0 **ΤΟΤΕ ΑΡΧΗ**

M_O_A ← Σ_A / ΠΛΗΘΟΣ_A

ΓΡΑΨΕ M_O_A

ΤΕΛΟΣ

ΤΕΛΟΣ

Μέγιστα - ελάχιστα

ΑΛΓΟΡΙΘΜΟΣ ΜΕΓ_ΕΛΑΧ

ΜΕΤΑΒΛΗΤΕΣ

X[10], max, min, i: **ΑΚΕΡΑΙΕΣ**

ΑΡΧΗ

ΓΙΑ i ← 1 **ΜΕΧΡΙ** 10 **ΑΡΧΗ**

ΔΙΑΒΑΣΕ X[i]

ΤΕΛΟΣ

max←X[1]

min←X[1]

//αρχικοποίηση στο πρώτο στοιχείο

ΓΙΑ i ← 1 **ΜΕΧΡΙ** 10 **ΑΡΧΗ**

ΑΝ X[i]>max **ΤΟΤΕ**

max←X[i]

ΑΝ X[i]<min **ΤΟΤΕ**

min←X[i]

ΤΕΛΟΣ

//σε αυτό το σημείο, η μέγιστη και η

//ελάχιστη τιμή έχουν υπολογιστεί

ΓΙΑ i ← 1 **ΜΕΧΡΙ** 10 **ΑΡΧΗ**

//διαπερνούμε από την αρχή τον πίνακα

ΑΝ X[i]=max **ΤΟΤΕ**

ΓΡΑΨΕ 'βρέθηκε μέγιστος στη θέση

' , i

ΑΝ X[i]=min **ΤΟΤΕ**

ΓΡΑΨΕ 'βρέθηκε ελάχιστος στη

θέση ' , i

ΤΕΛΟΣ

ΤΕΛΟΣ

```
#include <stdio.h>
```

```
int main(void){
```

```
int x[11], i, max, min;
```

```
for(i=1;i<=10;i++){
```

```
scanf("%d", &x[i]);flush(stdin);
```

```
}
```

```
max=x[1];min=x[1];
```

```
for(i=1;i<=10;i++){
```

```
if(x[i]>max)
```

```
max=x[i];
```

```
if (x[i]<min)
```

```
min=x[i];
```

```
}
```

```
for(i=1;i<=10;i++){
```

```
if(x[i]==max)
```

```
printf("max in position %d\n",
```

```
i);
```

```
if (x[i]==min)
```

```
printf("min in position %d\n",
```

```
i);;
```

```
}
```

```
getchar();
```

```
}
```

«Δίνεται μονοδιάστατος πίνακας 10 ακεραίων. Να βρεθεί εάν τα στοιχεία του είναι όλα θετικά ή όχι».

ΑΛΓΟΡΙΘΜΟΣ ΘΕΤΙΚΟΙ_ΑΡΙΘΜΟΙ

ΜΕΤΑΒΛΗΤΕΣ

X[10], i: **ΑΚΕΡΑΙΕΣ**

θετικοί: **ΛΟΓΙΚΕΣ**

ΑΡΧΗ

ΓΙΑ i ← 1 **ΜΕΧΡΙ** 10 **ΑΡΧΗ**

ΔΙΑΒΑΣΕ X[i]

ΤΕΛΟΣ

θετικοί ← **true**

//έστω ότι όλοι αριθμοί είναι θετικοί

i ← 1

ΟΣΟ ((i ≤ 10) **and** (θετικοί = **true**)) **ΑΡΧΗ**

ΑΝ X[i] < 0 **ΤΟΤΕ**

θετικοί ← **false**

//βρέθηκε ένας αρνητικός αριθμός

i ← i + 1

ΤΕΛΟΣ

//μετά το τέλος της επανάληψης η μεταβλητή

//θετικοί, περιέχει το αποτέλεσμα

ΑΝ θετικοί = **true** **ΤΟΤΕ**

ΓΡΑΨΕ 'όλοι οι αριθμοί είναι θετικοί'

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'ΔΕΝ είναι όλοι οι αριθμοί θετικοί'

ΤΕΛΟΣ

```
#include <stdio.h>
int main(void){
    int x[11], i, positive;
    for(i=1;i<=10;i++){
        scanf("%d", &x[i]);fflush(stdin);
    }
    positive=1;
    i=1;
    while((i<=10)&&(positive)){
        if(x[i]<0)
            positive=0;
        i++;
    }
    if(positive)
        printf("all are positives");
    else
        printf("at least one negative");
    getchar();
}
```

«Έστω ένα πίνακας 10 χαρακτήρων. Μέσα στον πίνακα αποθηκεύεται ένα αλφαριθμητικό ίδιου μεγέθους. Να βρεθεί εάν το αλφαριθμητικό είναι καρκινικό (παλίνδρομο), διαβάζεται δηλαδή το ίδιο με φορά προς τα δεξιά και με φορά προς τα αριστερά.»

ΑΛΓΟΡΙΘΜΟΣ ΘΕΤΙΚΟΙ_ΑΡΙΘΜΟΙ

ΜΕΤΑΒΛΗΤΕΣ

X[10]: ΧΑΡΑΚΤΗΡΕΣ

παλίνδρομο: ΛΟΓΙΚΕΣ

i: ΑΚΕΡΑΙΕΣ

ΑΡΧΗ

ΓΙΑ i ← 1 ΜΕΧΡΙ 10 ΑΡΧΗ

ΔΙΑΒΑΣΕ X[i]

ΤΕΛΟΣ

παλίνδρομο ← true //έστω ότι είναι καρκινικό

i ← 1

ΟΣΟ ((i ≤ 10 div 2) and (παλίνδρομο = true)) ΑΡΧΗ

ΑΝ X[i] <> X[10-i+1] ΤΟΤΕ

παλίνδρομο ← false

i ← i+1

ΤΕΛΟΣ

ΑΝ παλίνδρομο = true ΤΟΤΕ

ΓΡΑΨΕ 'το αλφαριθμητικό είναι παλίνδρομο'

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'το αλφαριθμητικό δεν είναι παλίνδρομο'

ΤΕΛΟΣ

«Δίνεται μονοδιάστατος πίνακας 15 ακεραίων. Να αντιστραφεί ο πίνακας (το 1^ο στοιχείο να γίνει τελευταίο, το 2^ο προτελευταίο κ.ο.κ)»

**ΑΛΓΟΡΙΘΜΟΣ ΑΝΤΙΣΤΡΟΦΗ
ΜΕΤΑΒΛΗΤΕΣ**

X[15], I, Temp: **ΑΚΕΡΑΙΕΣ**

ΑΡΧΗ

ΓΙΑ i ← 1 ΜΕΧΡΙ 15 ΑΡΧΗ

ΔΙΑΒΑΣΕ X[i]

ΤΕΛΟΣ

ΓΙΑ i ← 1 ΜΕΧΡΙ 15 div 2 ΑΡΧΗ

Temp ← X[i]

X[i] ← X[15-i+1]

X[15-i+1] ← Temp

ΤΕΛΟΣ

ΓΙΑ i ← 1 ΜΕΧΡΙ 15 ΑΡΧΗ

ΓΡΑΨΕ X[i]

ΤΕΛΟΣ

ΤΕΛΟΣ

«Δίνονται δύο μονοδιάστατοι πίνακες 10 ακεραίων. Να βρεθεί το γινόμενο τους.»

$$S=A[1]*B[1]+A[2]*B[2]+...+A[K]*B[K] = \sum_{i=1}^K A[i] * B[i]$$

```
ΑΛΓΟΡΙΘΜΟΣ ΓΙΝΟΜΕΝΟ
ΜΕΤΑΒΛΗΤΕΣ
    A[10], B[10], I, S: ΑΚΕΡΑΙΕΣ
ΑΡΧΗ
    ΓΙΑ i ← 1 ΜΕΧΡΙ 10 ΑΡΧΗ
        ΔΙΑΒΑΣΕ A[i], B[i]
    ΤΕΛΟΣ
    S ← 0
    ΓΙΑ i ← 1 ΜΕΧΡΙ 10 ΑΡΧΗ
        S ← S + A[i] * B[i]
    ΤΕΛΟΣ
    ΓΡΑΨΕ S
ΤΕΛΟΣ
```

