

Ενότητα 2

Πίνακες

ΗΥ240 - Πανεγιότητα Φατούρου

1

Πίνακες

Ένας **πίνακας** είναι μια δομή δεδομένων που αποθηκεύει μια ακολουθία από διαδοχικά αριθμημένα αντικείμενα ιδίου τύπου, η οποία μπορεί να συμπεριλάβει μέχρι ένα προκαθορισμένο μέγιστο αριθμό αντικειμένων.

Πιο φορμαλιστικά:

- Έστω s, u δύο ακέραιοι.
- Συμβολίζουμε με $s..u$ το σύνολο των ακεραίων j τ.ω. $s \leq j \leq u$ (δηλαδή το **διάστημα ακεραίων** $s..u$).
- Ένας πίνακας είναι μια συνάρτηση $1-1$ και επί με πεδίο ορισμού ένα πεπερασμένο διάστημα ακεραίων, που λέγεται **σύνολο δεικτών**, και πεδίο τιμών ένα σύνολο από αντικείμενα (τα στοιχεία του πίνακα).
- Αν ο A είναι ένας πίνακας και j είναι ένας ακέραιος (δείκτης) στο σύνολο δεικτών του A , συμβολίζουμε με $A[j]$ το στοιχείο του A που βρίσκεται στη θέση j .

ΗΥ240 - Πανεγιότητα Φατούρου

2

Λειτουργίες σε Πίνακες

- **Access**(A, j): επιστρέφει το στοιχείο $A[j]$
- **Length**(A): επιστρέφει το πλήθος των στοιχείων του A
- **Assign**(A, j, v): εκτελεί τη λειτουργία $A[j] = v$
- **Initialize**(A, a): Αρχικοποιεί κάθε στοιχείο του A στην τιμή v
- **Iterate**(A, f): Εφαρμόζει την συνάρτηση f σε κάθε στοιχείο του A , ξεκινώντας από εκείνο που δεικτοδοτείται από το μικρότερο δείκτη του συνόλου δεικτών και καταλήγοντας σε εκείνο που δεικτοδοτείται από το μεγαλύτερο δείκτη.

Πολυδιάστατοι Πίνακες

- Ένας **διδιάστατος** πίνακας ορίζεται ως ο μονοδιάστατος πίνακας του οποίου τα στοιχεία είναι μονοδιάστατοι πίνακες.

- Ένας d -διάστατος πίνακας ορίζεται ως ο μονοδιάστατος πίνακας του οποίου τα στοιχεία είναι $(d-1)$ -διάστατοι πίνακες.

		Πίνακας A				
		0	1	2	3	4
0						
1						
2						
3						
4						

$A[1,3]$

- Αν $\langle j_1, \dots, j_d \rangle$ είναι ένα διάνυσμα d ακεραίων που ανήκει στο σύνολο δεικτών ενός d -διάστατου πίνακα A , τότε $A[j_1, \dots, j_d]$ είναι το στοιχείο του πίνακα A που βρίσκεται στη θέση j_2 του $(d-1)$ -διάστατο υποπίνακα $A[j_1]$ (που αντιστοιχεί στη θέση j_1 του A), στη θέση j_3 στον $(d-2)$ -διάστατο πίνακα $A[j_1, j_2]$ (που αντιστοιχεί στη θέση j_2 του $A[j_1]$), ..., στη θέση j_d στο μονοδιάστατο πίνακα $A[j_1, j_2, \dots, j_{d-1}]$ (που αντιστοιχεί στη θέση j_{d-1} του $A[j_1, \dots, j_{d-2}]$).

Πίνακες στη C

- Στη C οι πίνακες δεικτοδοτούνται ξεκινώντας από το 0 και το όνομα ενός πίνακα είναι και δείκτης στο 1ο στοιχείο του.
- Αν συμβολίσουμε με b το μέγεθος ενός στοιχείου του πίνακα A, τότε το j-οστό στοιχείο του πίνακα είναι το A[j-1] και βρίσκεται στη διεύθυνση $A + b^*(j-1)$.
- Κάποιες φορές το πλήθος των θέσεων του πίνακα που είναι κατειλημμένες από έγκυρα στοιχεία δεν είναι γνωστό και αντί αυτού χρησιμοποιείται ένα στοιχείο φρουρός το οποίο σηματοδοτεί το τέλος του πίνακα (αυτό π.χ. γίνεται στη C με τα αλφαριθμητικά).

Διδιάστατοι Πίνακες

Στη C, ένας διδιάστατος πίνακας A με n γραμμές και m στήλες αποθηκεύεται κατά γραμμές: A[0,0], A[0,1], ... , A[0, m], A[1,0], ... , A[1,m], ... , A[n,0], ... , A[n,m].

Άρα το στοιχείο A[k,j] βρίσκεται στη θέση μνήμης $A + b^*(m^*k + j)$. **Γιατί?**

	0	1	...	m
0	[0,0]	[0,1]	...	[0,m]
1	[1,0]	[1,1]	...	[1,m]
...	...	...	...	...
n	[n,0]	[n,1]	...	[n,m]

Συμμετρικοί Πίνακες

- Ένας διδιάστατος πίνακας με ίδιο πλήθος γραμμών και στηλών ονομάζεται **Τετραγωνικός**.
- **Συμμετρικός** ονομάζεται ένας τετραγωνικός πίνακας A για τον οποίο ισχύει πως $A[k,j] = A[j,k]$, για κάθε $0 \leq j,k \leq n-1$ (όπου n το πλήθος των γραμμών ή των στηλών του A).

	0	1	2
0	5	34	31
1	34	6	87
2	31	87	45
	Πίνακας A		

- Υπάρχουν $1 + 2 + \dots + n$ διαφορετικά στοιχεία και άρα απαιτούνται $(1+n)^*n/2$ θέσεις μνήμης για την αποθήκευση του πίνακα:

□ Τα στοιχεία του διδιάστατου πίνακα αποθηκεύονται σε έναν μονοδιάστατο πίνακα ως εξής:

0	1	2	3	4	5
5	34	6	31	87	45

Πως υπολογίζεται η Access(A,j₁,j₂) που επιστρέφει το στοιχείο A[j₁,j₂] του συμμετρικού πίνακα A;

Τριγωνικοί Πίνακες

- Άνω (κάτω) τριγωνικός ονομάζεται ένας τετραγωνικός πίνακας A για τον οποίο ισχύει πως, για κάθε $0 \leq j, k \leq n-1$, $A[k, j] = 0$ αν $k < j$ (ή $k > j$, αντίστοιχα), όπου n το πλήθος των γραμμών ή των στηλών του A .

Παράδειγμα Άνω Τριγωνικού Πίνακα

	0	1	2	3
0	1	0	0	0
1	21	43	0	0
2	34	7	18	0
3	5	86	23	43

Πως μπορεί να υλοποιηθεί ένας τριγωνικός πίνακας χωρίς μεγάλη σπατάλη μνήμης;

Πως υλοποιείται η $Access(A[i_1, j_2])$ που επιστρέφει το στοιχείο $A[i_1, j_2]$ ενός άνω (ή κάτω) τριγωνικού πίνακα A ;

Αραιοί Πίνακες

Παράδειγμα Αραιού Πίνακα A με 4 γραμμές και 5 στήλες

	0	1	2	3	4
0	0	0	7	0	0
1	1	1	2	0	0
2	0	0	0	4	0
3	12	0	0	0	0

Τρόπος Αποθήκευσης

Αποθήκευση ενός δυαδικού πίνακα B με m γραμμές και n στήλες για τον οποίο ισχύει πως $B[i, j] = 0$, αν $A[i, j] = 0$ και $B[i, j] = 1$, διαφορετικά, για κάθε $0 \leq i \leq n-1$ και $0 \leq j \leq m-1$, καθώς και ενός μονοδιάστατο πίνακα που θα περιέχει όλα τα μη-μηδενικά στοιχεία του A με τη σειρά που συναντώνται σε αυτόν (σύμφωνα με τον τρόπο αποθήκευσης των διδιόστατων πινάκων στη C).

Παράδειγμα: Ο πίνακας B που αντιστοιχεί στον Ο μονοδιάστατος πίνακας είναι: πίνακα A του παραπάνω παραδείγματος είναι:

	0	1	2	3	4
0	0	1	0	0	0
1	1	1	0	0	1
2	0	0	1	0	0
3	1	0	0	0	0

7	1	2	-3	4	12
---	---	---	----	---	----

Τι κερδίζουμε με αυτό τον τρόπο υλοποίησης του αραιού πίνακα;