ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΣΤΟΥΣ ΠΙΝΑΚΕΣ – ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ

|  |  |  |  |
| --- | --- | --- | --- |
| 23 | 4 | 56 | 2 |

Το Π[i] είναι το στοιχείο του πίνακα που βρίσκεται στη i θέση, για παράδειγμα στον διπλανό πίνακα το Π[2] είναι ο αριθμός 4.

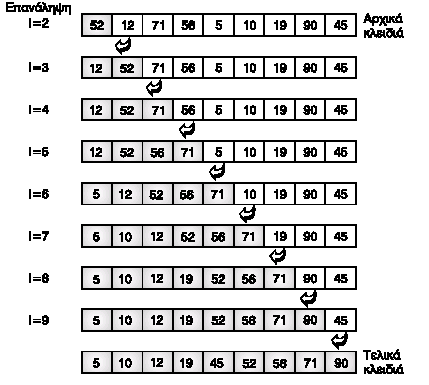
|  |  |
| --- | --- |
| Διάβασμα ενός μονοδιάστατου πίνακα Π, N ακεραίων:  Ψευδογλώσσα  ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΒΑΣΜΑ\_ΠΙΝΑΚΑ  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΔΙΑΒΑΣΕ Π[i]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ ΔΙΑΒΑΣΜΑ\_ΠΙΝΑΚΑ | Πρόγραμμα σε ΓΛΩΣΣΑ  ΠΡΟΓΡΑΜΜΑ ΔΙΑΒΑΣΜΑ\_ΠΙΝΑΚΑ  ΣΤΑΘΕΡΕΣ  Ν=100 ! Έστω ο πίνακας έχει 100 στοιχεία  ΜΕΤΑΒΛΗΤΕΣ  ΑΚΕΡΑΙΕΣ: Π[Ν], i  ΑΡΧΗ  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΔΙΑΒΑΣΕ Π[i]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ |
| Εμφάνιση ενός μονοδιάστατου πίνακα:  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΡΑΨΕ Π[i] **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** Μηδενισμός στοιχείων μονοδιάστατου πίνακα  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  Π[i]🡨0  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | Εύρεση αθροίσματος ενός μονοδιάστατου πίνακα:  ΑΘΡΟΙΣΜΑ←0  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΑΘΡΟΙΣΜΑ **←** ΑΘΡΟΙΣΜΑ + Π[i]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  Τον μέσο όρο μπορούμε στην συνέχεια να τον βρούμε με την εντολή  ΜΕΣΟΣ\_ΟΡΟΣ **←** ΑΘΡΟΙΣΜΑ / Ν |
| Εύρεση μέγιστου ενός μονοδιάστατου πίνακα:  ΜΕΓΙΣΤΟΣ **←** Π[1]  ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ Ν  ΑΝ Π[i] > ΜΕΓΙΣΤΟΣ ΤΟΤΕ  ΜΕΓΙΣΤΟΣ ← Π[i]  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  Αν θέλω να βρω ποια στοιχεία είναι ίσα με τον μέγιστο (σε περίπτωση ισοβαθμιών)  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΑΝ Π[i] = ΜΕΓΙΣΤΟΣ ΤΟΤΕ  ΓΡΑΨΕ i  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | Εύρεση μέγιστου ενός μονοδιάστατου πίνακα αλλά και της θέσης του:  ΜΕΓΙΣΤΟΣ ← Π[1]  ΘΕΣΗ ← 1  ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ Ν  ΑΝ Π[i] > ΜΕΓΙΣΤΟΣ ΤΟΤΕ  ΜΕΓΙΣΤΟΣ ← Π[i]  ΘΕΣΗ ← i  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ |
| Εμφάνιση στοιχείων ενός μονοδιάστατου πίνακα που επαληθεύουν μία συνθήκη (π.χ. αυτών που είναι μεγαλύτερα από 10):  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΑΝ Π[i] > 10 ΤΟΤΕ  ΓΡΑΨΕ Π[i]  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | Μέσος όρος στοιχείων ενός μονοδιάστατου πίνακα που επαληθεύουν μία συνθήκη π.χ. αυτών που είναι μεγαλύτερα από 10 και μικρότερα από 30.  Σ🡨0 !Χρειάζεται ένας αθροιστής και ένας μετρητής  μ🡨0  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΑΝ Π[i] > 10 ΚΑΙ Π[i] < 30 ΤΟΤΕ  Σ🡨Σ+ Π[i]  μ🡨μ+1  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΑΝ μ>0 ΤΟΤΕ ΜΕΣΟΣ\_ΟΡΟΣ **←** Σ / μ |
| **Παράλληλοι πίνακες**  Αποθηκεύουμε σε δύο πίνακες τους τίτλους (Τ) τραγουδιών και τα αντίστοιχα έτη (ΕΤΟΣ) που τραγουδήθήκαν για πρώτη φορά. Στη συνέχεια βρίσκουμε το παλαιότερο (υποθέτουμε ότι είναι ένα) π.χ.   |  |  |  |  | | --- | --- | --- | --- | | Χαιρετίσματα | Όλα δικά σου | Χέρια ψηλά | Αχ Ελλάδα |      |  |  |  |  | | --- | --- | --- | --- | | 1985 | 1970 | 2010 | 1995 | | **ΓΙΑ** I **ΑΠΟ** 1 **ΜΕΧΡΙ** Ν     **ΔΙΑΒΑΣΕ** Τ**[**i**],** ΕΤΟΣ**[i]**  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  ΜΙΝ **🡨** ΕΤΟΣ**[**1**]**  ΘΕΣΗ **🡨** 1 **ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** Ν   **ΑΝ** ΕΤΟΣ**[**i**]** **<** ΜΙΝ **ΤΟΤΕ**      ΜΙΝ **🡨** ΕΤΟΣ**[**i**]**       ΘΕΣΗ **🡨** i    **ΤΕΛΟΣ\_ΑΝ**  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  **ΓΡΑΨΕ** 'ΠΑΛΑΙΟΤΕΡΟ ΤΡΑΓΟΥΔΙ '**,** Τ**[**ΘΕΣΗ**],**  ΜΙΝ |
| Αναζήτηση σε μονοδιάστατο πίνακα  **Αναζήτηση της θέσης όλων των στοιχείων,** αποθηκευμένων σε μονοδιάστατο πίνακα, που πληρούν κάποιο κριτήριο (π.χ. να βρεθούν οι θέσεις όλων των 'Γιώργος', αν υπάρχουν) και εμφάνιση μηνύματος αποτυχίας αν δεν βρεθεί κανένα.  !Η «ΒΡΕΘΗΚΕ» είναι λογική μεταβλητή (σημαία).  ΔΙΑΒΑΣΕ key !η μεταβλητή key είναι το στοιχείο  !που αναζητούμε  ΒΡΕΘΗΚΕ ← ΨΕΥΔΗΣ  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΑΝ Π[i] = key ΤΟΤΕ  ΓΡΑΨΕ i  ΒΡΕΘΗΚΕ ← ΑΛΗΘΗΣ **ΤΕΛΟΣ\_ΑΝ** ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΑΝ ΒΡΕΘΗΚΕ = ΨΕΥΔΗΣ ΤΟΤΕ  ΓΡΑΨΕ ‘ΔΕΝ ΒΡΕΘΗΚΕ’  ΤΕΛΟΣ\_ΑΝ | **Αναζήτηση της θέσης του πρώτου στοιχείου** του πίνακα που πληροί κάποιο κριτήριο (π.χ. να βρεθεί η θέση του πρώτου «Γιώργος», αν υπάρχει) και εμφάνιση μηνύματος αποτυχίας αν δεν βρεθεί κανένα.  ΔΙΑΒΑΣΕ key  i ← 1  βρέθηκε ← ΨΕΥΔΗΣ  θέση🡨0 !η μεταβλητή «θέση» αντιστοιχεί στη θέση  !του πίνακα που πιθανόν θα βρεθεί το στοιχείο που ψάχνουμε  ΟΣΟ (i <= Ν) ΚΑΙ (βρέθηκε =ΨΕΥΔΗΣ) ΕΠΑΝΑΛΑΒΕ  ΑΝ Π[i]=key ΤΟΤΕ  βρέθηκε🡨ΑΛΗΘΗΣ  θέση🡨i  ΑΛΛΙΩΣ  i ← i + 1  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΑΝ βρέθηκε = ΑΛΗΘΗΣ ΤΟΤΕ ! ή Αν θέση<>0 τότε  ΓΡΑΨΕ θέση  ΑΛΛΙΩΣ  ΓΡΑΨΕ ‘ΔΕΝ ΒΡΕΘΗΚΕ’  ΤΕΛΟΣ\_ΑΝ |
| ΣΕΙΡΙΑΚΗ\_ΑΝΑΖΗΣΗ\_ΣΕ\_ΤΑΞΙΝΟΜΗΜΕΝΟ\_ΠΙΝΑΚΑ  ! Έστω ότι ψάχνουμε να βρούμε το κλειδί 'ΓΙΩΡΓΟΣ' σε ταξινομημένο κατά αύξουσα σειρά πίνακα ονομάτων. Ο αλγόριθμος σταματά όταν το βρει.  ΔΙΑΒΑΣΕ key  i ← 1  σταματα🡨ΨΕΥΔΗΣ  ΟΣΟ (i <= Ν) ΚΑΙ (σταματά=ΨΕΥΔΗΣ) ΕΠΑΝΕΛΑΒΕ  ΑΝ key>Π[i] ΤΟΤΕ  i ← i + 1  ΑΛΛΙΩΣ  σταμάτα🡨ΑΛΗΘΗΣ  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΑΝ σταμάτα = ΨΕΥΔΗΣ ΤΟΤΕ  ΓΡΑΨΕ ‘ΔΕΝ ΒΡΕΘΗΚΕ’  ΑΛΛΙΩΣ  ΑΝ key=Π[i] ΤΌΤΕ  ΓΡΑΨΕ 'βρέθηκε στη θέση ', i  ΤΕΛΟΣ\_ΑΝ **ΤΕΛΟΣ\_ΑΝ** | **Δυαδική Αναζήτηση**  Ο αλγόριθμος της δυαδικής αναζήτησης (binary search) εφαρμόζεται μόνο σε πίνακες που έχουν ταξινομημένα στοιχεία. Αν τα στοιχεία δεν είναι ταξινομημένα τότε δεν μπορεί να εφαρμοστεί. Στον αλγόριθμο αυτό, δεν συγκρίνουμε διαδοχικά κάθε στοιχείο του πίνακα με το προς αναζήτηση στοιχείο, όπως γίνεται στη σειριακή αναζήτηση.  Ο αλγόριθμος λειτουργεί ως εξής:  Βρίσκουμε το μεσαίο στοιχείο του ταξινομημένου πίνακα. Εάν το προς αναζήτηση στοιχείο είναι ίσο με το μεσαίο στοιχείο τότε σταματάμε την αναζήτηση αφού το στοιχείο βρέθηκε  Εάν δεν βρέθηκε, τότε ελέγχουμε αν το στοιχείο που αναζητούμε είναι μικρότερο ή μεγαλύτερο από το μεσαίο στοιχείο του πίνακα. Αν είναι μικρότερο, περιορίζουμε την αναζήτηση στο πρώτο μισό του πίνακα (με την προϋπόθεση ότι τα στοιχεία είναι διατεταγμένα κατά αύξουσα σειρά), ενώ αν είναι μεγαλύτερο περιορίζουμε την αναζήτηση στο δεύτερο μισό του πίνακα.  Η διαδικασία αυτή λοιπόν επαναλαμβάνεται για το κατάλληλο πρώτο ή δεύτερο μισό πίνακα, μετά για το 1/4 του πίνακα κ.ο.κ. μέχρι, είτε να βρεθεί το στοιχείο, είτε να μην είναι δυνατό να χωρισθεί ο πίνακας περαιτέρω σε δύο νέα μέρη. |
| **Δυαδική\_αναζήτηση**  !Π μονοδιάστατος πίνακας Ν θέσεων, key το στοιχείο που αναζητούμε  **δεδομένα //** N, Π, key //    Left 🡨 1 ! αριστερό όριο  Right 🡨 N ! δεξιό όριο  K 🡨 0 ! θέση του στοιχείου  F 🡨 FALSE  **όσο** (Left<=Right) και (f=FALSE) **επανάλαβε**  M 🡨 (Left+Right) div 2  **αν** Π[M]=key **τότε**  K🡨 M;  F 🡨 TRUE;  **αλλιώς**  **αν** Π[M]<key **τότε**  Left 🡨 M+1;  **αλλιώς**  Right🡨 M-1;  **Τέλος\_αν**  **Τέλος\_αν**  **Τέλος\_επανάληψης**  **Αν** F= TRUE **τότε**  **Εμφάνισε** "Το στοιχείο,", key , "υπάρχει στη θέση:", Μ  **Αλλιώς**  **Εμφάνισε "**Το στοιχείο,", key , " δεν υπάρχει στον πίνακα"  **Τέλος\_αν** | **Αλγόριθμος** Ταξινόμηση\_Φυσαλίδα !Ευθεία Ανταλλαγή  **Δεδομένα** // table, n //  **Για** i **από** 2 **μέχρι** n  **Για** j **από** n **μέχρι** i **με\_βήμα** –1  **Αν** table[j-1] > table[j] **τότε**  **αντιμετάθεσε** table[j-1], table[j]  **Τέλος\_αν**  **Τέλος\_επανάληψης**  **Τέλος\_επανάληψης**  **Αποτελέσματα** // table //  **Τέλος** Φυσαλίδα  Η αντιμετάθεσε πραγματοποιείται ως εξής:  temp 🡨 table[j-1]  table[j-1] 🡨 table[j]  table[j] 🡨 temp |
| Μία **βελτίωση στον αλγόριθμο της φυσαλίδας** είναι να σταματά όταν σε ένα πέρασμα δεν γίνεται καμία αντιμετάθεση στοιχείων. Το γεγονός αυτό σημαίνει ότι δεν υπάρχει μικρότερο στοιχείο να ανέβει προς τα πάνω και άρα όλα τα στοιχεία βρίσκονται στη σωστή θέση ή με άλλα λόγια ο πίνακας είναι ταξινομημένος. Ο βελτιστοποιημένος αλγόριθμος υλοποιημένος σε πρόγραμμα δίνεται παρακάτω:  **Πρόγραμμα** ΦΥΣΑΛΙΔΑ2 Σταθερές N=6 Μεταβλητές **ακέραιες**: προσωρινή, i, j, table [n]  **λογικές**: ταξινομημένος  !η μεταβλητή " ταξινομημένος " είναι αληθής όταν δεν γίνεται  !αντιμετάθεση και άρα ο πίνακας είναι ταξινομημένος  **Αρχή**  **Για** i **από** 1 **μέχρι** n !Διάβασμα του πίνακα  **Διάβασε** table [i]  **Τέλος\_επανάληψης** | ταξινομημένος 🡨 Ψευδής  i 🡨 2  **Όσο** i <= n **και** ταξινόμηση = Ψευδής **επανάλαβε**  ταξινομημένος 🡨 Αληθής !Έστω πίνακας ταξινομημένος  **Για** j **από** n **μέχρι** i **με\_βήμα** –1  **Αν** table [j] < table [j-1] **τότε**  προσωρινή 🡨table [j]  table [j] 🡨 table [j-1]  table [j-1] 🡨 προσωρινή  ταξινομημένος 🡨 Ψευδής !ο πίνακας δεν έχει  **Τέλος\_αν**  !ταξινομηθεί  **Τέλος\_επανάληψης**  i 🡨 i+1  **Τέλος\_επανάληψης**  **Για** i **από** 1 **μέχρι** 6  **Γράψε** table [i]  **Τέλος\_επανάληψης**  **Τέλος\_προγράμματος** |

**Ταξινόμηση Ευθείας Εισαγωγής**

Ο αλγόριθμος *ταξινόμιση ευθείας εισαγωγής* (straight insertion sort) είναι ιδανικός για περιπτώσεις δεδομένων που είναι «περίπου» ταξινομημένα και χρησιμοποιείται σε πολλά υβριδικά σχήματα.

Σύμφωνα με τον αλγόριθμο αυτό τα στοιχεία διακρίνονται σχηματικά σε μία *ακολουθία προορισμού* (destination sequence) table[1], table[2], ..., table[i-1] και σε μία *ακολουθία πηγής* (source sequence) table[i], ..., table[n]. Αρχικά η ακολουθία προορισμού αποτελείται από ένα στοιχείο, το πρώτο, και σταδιακά μεγαλώνει κατά ένα. Αυτό επιτυγχάνεται θεωρώντας το πρώτο στοιχείο της ακολουθίας πηγής και παρεμβάλλοντας το στην κατάλληλη θέση μεταξύ των στοιχείων της ακολουθίας προορισμού εκτελώντας διαδοχικές συγκρίσεις από τα δεξιά προς τα αριστερά με τα στοιχεία της ακολουθίας προορισμού. Ο σχετικός αλγόριθμος δίνεται στη συνέχεια.

**Παράδειγμα.** ‘Έστω ότι ο αρχικός πίνακας αποτελείται από τα εννέα κλειδιά 52, 12, 71, 56, 5, 10, 19, 90 και 45. Στο προηγούμενο σχήμα παρουσιάζεται η διαδικασία ταξινόμησης θεωρώντας τον πίνακα αυτό. Κάθε φορά η ακολουθία προορισμού εμφανίζεται με σκίαση, ενώ το πρώτο στοιχείο της ακολουθίας πηγής αναδεικνύεται από το αντίστοιχο βέλος. Το στοιχείο αυτό λαμβάνει την κατάλληλη θέση μέσα στην ακολουθία προορισμού “σπρώχνοντας” μερικά στοιχεία προς τα δεξιά. Η εύρεση της κατάλληλης θέσης γίνεται εύκολα με διαδοχικές συγκρίσεις και μετακινήσεις. Λόγου χάριν, στην πέμπτη σειρά το στοιχείο 10 συγκρίνεται διαδοχικά με τα στοιχεία 71, 56, 52, 12 και 5, οπότε γίνεται αντιληπτό ότι το 10 πρέπει να παρεμβληθεί μεταξύ των 5 και 12. Για να γίνει αυτό τα στοιχεία 12 ως και 71 μετακινούνται μία θέση προς τα δεξιά για να δημιουργηθεί μία κενή θέση για το 10.



**Αλγόριθμος** Ευθεία\_Εισαγωγή

**Δεδομένα** \\ table, n \\

**Για** i **από** 2 **μέχρι** n

temp ← table[i]

j ← i-1

**Οσο(** j>=1) **ΚΑΙ** (temp<table[j]) **επανάλαβε**

table[j+1] ← table[j]

j ← j-1

**Τέλος\_επανάληψης**

table[j+1] ← temp

**Τέλος\_επανάληψης**

**Αποτελέσματα** \\ table \\

**Τέλος** Ευθεία\_Εισαγωγή

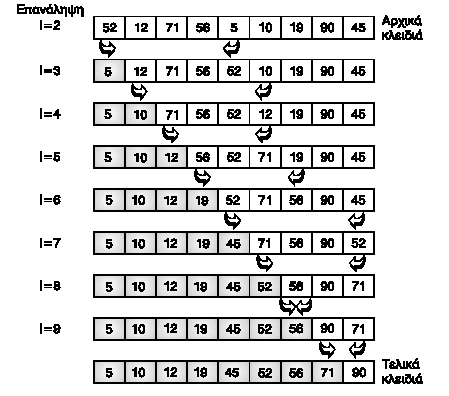
**Ταξινόμηση ευθείας επιλογής**

Η μέθοδος αυτή βασίζεται στις ακόλουθες δύο αρχές:

* επιλογή του στοιχείου με το ελάχιστο κλειδί και
* ανταλλαγή αυτού του στοιχείου με το πρώτο στοιχείο του πίνακα.
* Αυτές οι λειτουργίες επαναλαμβάνονται για τα υπόλοιπα n-1 στοιχεία, μέχρι στο τέλος να απομείνει μόνο το μεγαλύτερο στοιχείο.

Η μέθοδος που περιγράφηκε προηγουμένως ονομάζεται ***ταξινόμηση ευθείας επιλογής*** (straight selection sort) και μπορεί να θεωρηθεί ως αντίθετη της ταξινόμησης ευθείας εισαγωγής. Πιο συγκεκριμένα, η ευθεία εισαγωγή θεωρεί σε κάθε βήμα αφ’ ενός το ένα και μοναδικό επόμενο στοιχείο της ακολουθίας πηγής και αφ’ ετέρου όλα τα στοιχεία της ακολουθίας προορισμού για να εντοπίσει το κατάλληλο σημείο εισαγωγής. Αντίθετα, η ευθεία επιλογή θεωρεί τα στοιχεία της ακολουθίας πηγής, ώστε να ανιχνεύσει το στοιχείο με το ελάχιστο κλειδί και να το τοποθετήσει ως το επόμενο στοιχείο της ακολουθίας προορισμού.

**Παράδειγμα.** Η μέθοδος αυτή παρουσιάζεται στο επόμενο σχήμα καθώς εφαρμόζεται στα ίδια γνωστά εννέα κλειδιά. Και πάλι, το ταξινομημένο τμήμα του πίνακα εμφανίζεται με σκίαση, ενώ με τα βέλη εμφανίζονται τα στοιχεία που ανταλλάσσονται αμοιβαία. Λόγου χάριν, στην πρώτη σειρά βρίσκουμε ότι το στοιχεί ο 5 είναι το μικρότερο και αντιμετατίθεται με το πρώτο στοιχείο του πίνακα, το 52. Έτσι προκύπτει η μορφή του πίνακα στη δεύτερη σειρά. Στη συνέχεια η διαδικασία προχωρεί με την ίδια λογική μέχρι την τελική ταξινόμηση του πίνακα.

******

Ο αλγόριθμος για την υλοποίηση της μεθόδου ταξινόμησης ευθείας επιλογής δίνεται παρακάτω.

**Αλγόριθμος** Ευθεία\_Επιλογή

**Δεδομένα** \\ table, n \\

**Για** i **από** 1 **μέχρι** n-1

min ← table[i]

k ← i

**Για** j **από** i+1 **μέχρι** n

**Αν** table[j]<min **τότε**

min🡨 table[j]

k🡨j

**Τέλος\_αν**

**Τέλος\_επανάληψης**

! τοποθέτηση του min στη σωστή θέση.

table[k]🡨 table[i]

table[i] 🡨 min

**Τέλος\_επανάληψης**

**Αποτελέσματα** \\ table \\

**Τέλος** Ευθεία\_ Επιλογή

*Για την ταξινόμηση δεδομένων έχουν εκπονηθεί πάρα πολλοί αλγόριθμοι. Παραπάνω παρουσιάστηκαν τρεις απλοί αλγόριθμοι ταξινόμησης. Από τους καλύτερους αλγορίθμους ταξινόμησης είναι η “γρήγορη ταξινόμηση” (quicksort). Η ταξινόμηση φυσαλίδας είναι ο πιο απλός και ταυτόχρονα από τους πιο αργούς αλγόριθμους ταξινόμησης.*

ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΣΤΟΥΣ ΠΙΝΑΚΕΣ – ΔΙΣΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ

Το Π[i,j] είναι το στοιχείο του πίνακα που βρίσκεται στη i γραμμή και τη j στήλη, για παράδειγμα στον παρακάτω πίνακα το Π[1,2] είναι η λέξη ‘γάτα’.

|  |  |  |
| --- | --- | --- |
| σκυλί | γάτα | πεταλούδα |
| άλογο | πρόβατο | αγελάδα |

|  |  |
| --- | --- |
| Διάβασμα ενός δισδιάστατου πίνακα ακεραίων NxM:  ψευδογλώσσα  ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΒΑΣΜΑ\_ΔΙΣΔ  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Μ  ΔΙΑΒΑΣΕ Π[i, j]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ ΔΙΑΒΑΣΜΑ\_ΔΙΣΔ | Πρόγραμμα σε ΓΛΩΣΣΑ  ΠΡΟΓΡΑΜΜΑ ΔΙΑΒΑΣΜΑ\_ΔΙΣΔ  ΣΤΑΘΕΡΕΣ  Ν=10  Μ=5  ΜΕΤΑΒΛΗΤΕΣ  ΑΚΕΡΑΙΕΣ: Π[Ν, Μ], i, j  ΑΡΧΗ  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Μ  ΔΙΑΒΑΣΕ Π[i, j]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ |
| Εμφάνιση ενός δισδιάστατου πίνακα N x M:  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Μ  ΓΡΑΨΕ Π[i, j]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | Εύρεση του αθροίσματος των στοιχείων ενός δισδιάστατου πίνακα N x M:  ΑΘΡΟΙΣΜΑ ← 0  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Μ  ΑΘΡΟΙΣΜΑ ← ΑΘΡΟΙΣΜΑ + Π[i, j]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  Τον μέσο όρο μπορούμε στην συνέχεια να τον βρούμε με την εντολή  ΜΕΣΟΣ\_ΟΡΟΣ ← ΑΘΡΟΙΣΜΑ / ( Ν \* Μ) |
| Εμφάνιση μερικών στοιχείων ενός δισδιάστατου πίνακα (π.χ. αυτών που είναι μεγαλύτερα από 10):  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Μ  ΑΝ Π[i, j]>10 ΤΟΤΕ  ΓΡΑΨΕ Π[i, j]  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | Για να υπολογίσουμε π.χ. το άθροισμα μίας γραμμής (ή μίας στήλης) ενός δισδιάστατου πίνακα (Ν x Μ), δεν χρειάζονται Ν x Μ επαναλήψεις, αλλά μόνο Μ (ή Ν). Αυτό σημαίνει ότι για μια μόνο γραμμή θέλουμε μόνο ένα *ΓΙΑ* και όχι *ΓΙΑ* μέσα σε *ΓΙΑ*.  Π.χ. υπολογισμός αθροίσματος της 5ης γραμμής ενός πίνακα:  ΑΘΡΟΙΣΜΑ ← 0  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Μ  ΑΘΡΟΙΣΜΑ ← ΑΘΡΟΙΣΜΑ + Π[5, j]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ |
| Εύρεση ελάχιστου ενός δισδιάστατου πίνακα N x M:  ΕΛΑΧΙΣΤΟ ← Π[1, 1]  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Μ  ΑΝ Π[i, j] < ΕΛΑΧΙΣΤΟ ΤΟΤΕ  ΕΛΑΧΙΣΤΟ← Π[i, j]  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  \*Δεν είναι δυνατόν να κάνουμε  ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ Ν γιατί χάνουμε όλη την πρώτη γραμμή.  \*Αρχικοποίηση στους πίνακες δεν χρειάζεται να είναι κάποια αυθαίρετη τιμή που πρέπει να σκεφτούμε αλλά το πρώτο στοιχείο του πίνακα. | Εύρεση ελάχιστου ενός δισδιάστατου πίνακα N x M, αλλά και της θέσης του:  ΕΛΑΧΙΣΤΟ ← Π[1, 1]  Θi ← 1  Θj ← 1  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Μ  ΑΝ Π[i, j] < ΕΛΑΧΙΣΤΟ ΤΟΤΕ  ΕΛΑΧΙΣΤΟ 🡨 Π[i, j]  Θi ← i  Θj ← j  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ |
| Αναζήτηση σε δισδιάστατο πίνακα  Αναζήτηση της θέσης **όλων** των στοιχείων ενός δισδιάστατου πίνακα που πληρούν κάποια προϋπόθεση (π.χ. να βρεθούν οι θέσεις όλων των "Γιώργος", αν υπάρχουν) και να εμφανιστεί μήνυμα αποτυχίας, αν δεν βρεθεί κανένα.  ΔΙΑΒΑΣΕ key  βρέθηκε ← ΨΕΥΔΗΣ  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Μ  ΑΝ Π[i, j] = keyΤΟΤΕ  ΓΡΑΨΕ i, j  βρέθηκε ← ΑΛΗΘΗΣ  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΑΝ βρέθηκε = ΨΕΥΔΗΣ  ΓΡΑΨΕ "ΔΕΝ ΒΡΕΘΗΚΕ"  ΤΕΛΟΣ\_ΑΝ | **Αναζήτηση** της θέσης του **πρώτου** στοιχείου ενός δισδιάστατου πίνακα που πληροί κάποια προϋπόθεση (π.χ. να βρεθεί η θέση του πρώτου Γιώργος, αν υπάρχει) και μηνύματος αποτυχίας, αν δεν βρεθεί κανένα.  ΔΙΑΒΑΣΕ key  i ← 1  βρέθηκε 🡨 ΨΕΥΔΗΣ  ΟΣΟ (i <= Ν) ΚΑΙ (βρέθηκε = ψευδής) ΕΠΑΝΕΛΑΒΕ  j ← 1  ΟΣΟ (j <= Μ) ΚΑΙ (βρέθηκε = ψευδής) ΕΠΑΝΕΛΑΒΕ  Αν Π[i,j] = key τότε  βρέθηκε🡨ΑΛΗΘΗΣ  γραμμή🡨i  στήλη 🡨j  Αλλιώς  j ← j + 1  Τέλος\_αν  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  i ← i + 1  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΑΝ βρέθηκε= ΑΛΗΘΗΣ ΤΟΤΕ  ΓΡΑΨΕ γραμμή, στήλη  ΑΛΛΙΩΣ  ΓΡΑΨΕ "ΔΕΝ ΒΡΕΘΗΚΕ"  ΤΕΛΟΣ\_ΑΝ |
| **Διάβασμα δισδιάστατου πίνακα κατά γραμμή**    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M  ΔΙΑΒΑΣΕ Π[i, j]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  **Διάβασμα δισδιάστατου πίνακα κατά στήλη**  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N  ΔΙΑΒΑΣΕ Π[i, j]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | **Άθροισμα στοιχείων δισδιάστατου** NxM **πίνακα κατά γραμμή**  **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** Ν  ΑΘΡΟΙΣΜΑ **←** 0  **ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** Μ  ΑΘΡΟΙΣΜΑ **←** ΑΘΡΟΙΣΜΑ + Π[i, j]  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  ΓΡΑΨΕ ΑΘΡΟΙΣΜΑ ! ΣΓ[i]🡨 ΑΘΡΟΙΣΜΑ  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ !**Αποθήκευση αθροισμάτων σε πίνακα |
| **Άθροισμα στοιχείων δισδιάστατου** NxM **πίνακα κατά στήλη**  **ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** M  ΑΘΡΟΙΣΜΑ **←** 0  **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** N  ΑΘΡΟΙΣΜΑ **←** ΑΘΡΟΙΣΜΑ + Π[i, j]  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  ΓΡΑΨΕ ΑΘΡΟΙΣΜΑ !ΣΣ[i]🡨 ΑΘΡΟΙΣΜΑ  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  |  |  |  | **🡪** |  | |  |  |  |  | **🡪** |  | |  |  |  |  | **🡪** |  | |  |  |  |  | **🡪** |  | | **↓** | **↓** | **↓** | **↓** |  | Αθροίσματα γραμμών ΣΓ | |  |  |  |  |  |  | | Αθροίσματα στηλών ΣΣ | | | |  |  | |
| **Άθροισμα στοιχείων δισδιάστατου** NxM **πίνακα κατά γραμμή αποθηκεύοντας τα αποτελέσματα κατευθείαν στον πίνακα ΣΓ**  **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** Ν  ΣΓ[i] **←** 0  **ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** Μ  ΣΓ[i] **←** ΣΓ[i] + Π[i, j]  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  ΓΡΑΨΕ ΣΓ[i]  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** | **Άθροισμα στοιχείων δισδιάστατου** NxM **πίνακα κατά στήλη αποθηκεύοντας τα αποτελέσματα κατευθείαν στον πίνακα ΣΣ**  **ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** M  ΣΣ[i] **←** 0  **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** N  ΣΣ[i] **←** ΣΣ[i] + Π[i, j]  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  ΓΡΑΨΕ ΣΣ[i]  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** |
| **Μέτρηση θετικών στοιχείων ανά γραμμή δισδιάστατου** NxM **πίνακα Π, αποθηκεύοντας τα αποτελέσματα κατευθείαν στον πίνακα ΜΓ**  **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** Ν  Μ **←** 0  **ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** Μ  **ΑΝ** Π[i, j]>0 **ΤΟΤΕ**  Μ**←** Μ+ 1  ΤΕΛΟΣ\_ΑΝ  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  ΓΡΑΨΕ Μ  ΜΓ[i]🡨Μ  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** | **Μέτρηση θετικών στοιχείων ανά στήλη δισδιάστατου** NxM **πίνακα Π, αποθηκεύοντας τα αποτελέσματα κατευθείαν στον πίνακα ΜΣ**  **ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** M  Μ **←** 0  **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** N  **ΑΝ** Π[i, j]>0 **ΤΟΤΕ**  Μ**←** Μ+ 1  ΤΕΛΟΣ\_ΑΝ  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  ΓΡΑΨΕ Μ  ΜΣ[j]🡨Μ  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** |
| **Άθροισμα στοιχείων δισδιάστατου** NxM **πίνακα κατά γραμμή και στήλη σε μία επανάληψη.**  **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** Ν  ΣΓ[i] **←** 0  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** Μ  ΣΣ[i] **←** 0  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** Ν  **ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** Μ  ΣΓ[i] **←** ΣΓ[i] + Π[i, j]  ΣΣ[j] **←** ΣΣ[j] + Π[i, j]  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** | **Εύρεση σε δισδιάστατο** NxM **πίνακα του μικρότερου στοιχείου κάθε γραμμής και της θέσης του**  **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** Ν  ΕΛΑΧΙΣΤΟ **←** Π[i, 1] !κάθε γραμμή σαν μονοδιάστατος πίνακας  ΘΕΣΗ🡨1  **ΓΙΑ** j **ΑΠΟ** 2 **ΜΕΧΡΙ** Μ  ΑΝ Π[i, j] < ΕΛΑΧΙΣΤΟ ΤΟΤΕ  ΕΛΑΧΙΣΤΟ 🡨 Π[i, j]  ΘΕΣΗ🡨j  ΤΕΛΟΣ\_ΑΝ  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  ΓΡΑΨΕ ΕΛΑΧΙΣΤΟ, ΘΕΣΗ ! ΕΓ[i]🡨 ΕΛΑΧΙΣΤΟ  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ !**Αποθήκευση ελαχίστων σε πίνακα |
| **Εύρεση σε δισδιάστατο** NxM **πίνακα του μικρότερου στοιχείου κάθε στήλης και της θέσης του**  **ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** M  ΕΛΑΧΙΣΤΟ **←** Π[1, j] !κάθε στήλη σαν  ΘΕΣΗ🡨1 ! μονοδιάστατος πίνακας  **ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** N  ΑΝ Π[i, j] < ΕΛΑΧΙΣΤΟ ΤΟΤΕ  ΕΛΑΧΙΣΤΟ 🡨 Π[i, j]  ΘΕΣΗ🡨i  ΤΕΛΟΣ\_ΑΝ  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  ΓΡΑΨΕ ΕΛΑΧΙΣΤΟ, ΘΕΣΗ  ! ΕΣ[i]🡨 ΕΛΑΧΙΣΤΟ **!**Αποθήκευση ελαχίστων **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**    **Άθροισμα δύο ΝxΜ δισδιάστατων πινάκωνA,B σε Γ.**  **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** N  **ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** Μ  Γ[i,j]🡨 A[i,j]+ B[i,j]  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** | Να υπολογιστεί το **άθροισμα των δύο κύριων διαγωνίων** ενός τετραγωνικού πίνακα ΝxN.  Ας θεωρήσουμε ένα τετραγωνικό πίνακα 6x6 Α[6,6].   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 23 | 3 | 21 | 29 | 8 | 31 | | 9 | 27 | 91 | 12 | 29 | 109 | | 23 | 32 | 17 | 13 | 41 | 21 | | 34 | 21 | 5 | 14 | 40 | 17 | | 65 | 12 | 4 | 18 | 30 | 59 | | 78 | 1 | 23 | 21 | 29 | 64 |   Τα στοιχεία της πρώτης διαγωνίου είναι τα Α[1,1], Α[2,2], Α[3,3], Α[4,4], Α[5,5], Α[6,6]. Έχουν γενική μορφή δηλαδή **Α[i, i].** Το άθροισμα των παραπάνω στοιχείων είναι Σ1 🡨 Α[1,1] + Α[2,2] + Α[3,3] + Α[4,4] + Α[5,5] + Α[6,6].  Τα στοιχεία της δεύτερης διαγωνίου είναι τα Α[1,6], Α[2,5], Α[3,4], Α[4,3], Α[5,2], Α[6,1]. Έχουν γενική μορφή δηλαδή **Α[i, 7-i].** Το άθροισμα των παραπάνω στοιχείων είναι Σ2 🡨 Α[1,6] + Α[2,5] + Α[3,4] + Α[4,3] + Α[5,2] + Α[6,1].  Το άθροισμα και των δύο διαγωνίων είναι: Σ 🡨Σ1+Σ2  **Πρόγραμμα** Άθροισμα\_Διαγώνιων Μεταβλητές **Ακέραιοι**: i, j  **Πραγματικές**: Α[6,6], Σ1, Σ2, Σ  **Αρχή**  **Για** i **από** 1 **μέχρι** 6  **Για** j **από** 1 **μέχρι** 6  **Γράψε** ‘Δώσε το στοιχείο της ‘ ,i, ‘γραμμής και της ‘, j, ‘στήλης’  **Διάβασε** Α[i,j]  **Τέλος\_επανάληψης**  **Τέλος\_επανάληψης**  Σ1🡨0  Σ2🡨0  Σ🡨0  **Για** i **από** 1 **μέχρι** 6  Σ1🡨Σ1+ Α[i,i]  Σ2🡨Σ2+ Α[i,7-i] !A[i,N+1-i]  **Τέλος\_επανάληψης**  Σ🡨Σ1+ Σ2  **Γράψε** ‘Το άθροισμα των διαγώνιων στοιχείων είναι:’, Σ  **Τέλος\_προγράμματος** Άθροισμα\_Διαγώνιων |

**Λυμένες Ασκήσεις από το Τετράδιο Μαθητή - Κεφάλαιο 3 – Πίνακες**

**ΔΤ6**. Αποτελεί απλή παραλλαγή της μεθόδου ταξινόμησης ευθείας ανταλλαγής, που περιγράφεται στην παράγραφο 3.7. Η διαφορά έγκειται στο ότι στον εξωτερικό βρόχο δεν πρέπει να γίνουν n-1 αλλά 10 επαναλήψεις, ώστε να απομονωθούν στην κορυφή του πίνακα οι 10 μικρότερες ζητούμενες τιμές που θα είναι οι χώρες με τα μικρότερα ποσοστά κάλυψης της έκτασής τους με δάση.

Θα χρησιμοποιήσουμε ένα πίνακα με 50 ονόματα για τις χώρες ( Χώρες[50] ) και ένα πίνακα με 50 πραγματικούς αριθμούς για την αποθήκευση των ποσοστών των δασών ( Δάση[50]).

Ο αλγόριθμος αποτελείται από τρία μέρη:

1. Διάβασμα των δεδομένων
2. Ταξινόμηση του πίνακα Δάση για να βρούμε τις χώρες με τα μικρότερα ποσοστά.
3. Εκτύπωση των 10 χωρών με τα μικρότερα ποσοστά.

**Αλγόριθμος** Δάση

**Για** i **από** 1 **μέχρι** 50

**Εμφάνισε** 'Δώσε όνομα χώρας και ποσοστό της έκτασής της σε δάση'

**Διάβασε** Χώρες[i]

**Διάβασε** Δάση[i]

**Τέλος\_Επανάληψης**

**Για** i **από** 2 **μέχρι** 11

**Για** j **από** 50 **μέχρι** i **με\_βήμα** -1

**Αν** Δάση[j-1] > Δάση[j] **τότε**

προσωρινή1 <-- Δάση[j]

Δάση[j] <-- Δάση[j-1]

Δάση[j-1] <-- προσωρινή1

!Οι χώρες ακολουθούν στην αντιμετάθεση τα δάση, γιατί η 3η χώρα για παράδειγμα αντιστοιχεί στο 3ο δάσος, και αν το δάσος

!αλλάξει θέση λόγου αντιμετάθεσης πρέπει το ίδιο να γίνει και με τη χώρα.

προσωρινή2 <-- Χώρες[j]

Χώρες[j] <-- Χώρες[j-1]

Χώρες[j-1] <-- προσωρινή2

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

!εκτύπωση των 10 χωρών και των ποσοστών τους

**Για** i **από** 1 **μέχρι** 10

**Εμφάνισε** Χώρες[i], Δάση[i]

**Τέλος\_επανάληψης**

**Τέλος** Δάση

## ΔΣ2. Θα θεωρήσουμε ένα δισδιάστατο πίνακα scores μεγέθους 5x5, όπου κάθε γραμμή του πίνακα αντιστοιχεί σε ένα παίκτη, ενώ κάθε στήλη του πίνακα αντιστοιχεί σε έναν αγώνα. Τους συνολικούς πόντους που πέτυχαν οι παίκτες, τους αποθηκεύουμε σε ένα μονοδιάστατο πίνακα 5 θέσεων που ονομάζεται sum. Η μεταβλητή first δηλώνει τον αθλητή που επέτυχε συνολικά τους περισσότερους πόντους (max).

**Αλγόριθμος** Στατιστικά

**Δεδομένα** // sum, scores //

**Για** i **από** 1 **μέχρι** 5

sum[i] ← 0

**Τέλος\_επανάληψης**

**Για** i **από** 1 **μέχρι** 5

**Για** j **από** 1 **μέχρι** 5

sum[i] ← sum[i]+ scores[i,j]

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

max ← sum[1]

first ← 1

**Για** i **από** 2 **μέχρι** 5

**Αν** scores[i] > max **τότε**

max ← scores[i]

first ← i

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Αποτελέσματα** // first //

**Τέλος** Στατιστικά

**Λυμένες Ασκήσεις από το Τετράδιο Μαθητή - Κεφάλαιο 9 – Πίνακες**

**ΔΣ6. Eurobasket**

¨Έστω ότι έχουμε 5 ομάδες τα ονόματα των οποίων είναι αποθηκευμένα στον πίνακα Χώρες[5]. Οι πίνακες των αποτελεσμάτων Σ1 και των διαφορών πόντων Π1 μπορούν να έχουν την παρακάτω μορφή:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ν | Ν | Η | Ν |  |  |  | 23 | 14 | 7 | 2 |
|  |  | Ν | Η | Η |  |  |  |  | 23 | 5 | 6 |
|  |  |  | Ν | Ν |  |  |  |  |  | 3 | 8 |
|  |  |  |  | Ν |  |  |  |  |  |  | 4 |
|  |  |  |  |  |  |  |  |  |  |  |  |

Παρατηρούμε ότι οι πίνακες είναι τριγωνικοί αφού οι αγώνες γίνονται μία φορά. Παρατηρούμε ότι από τον πίνακα Σ1 η 1η ομάδα έχασε από την 4η και άρα η διαφορά πόντων πρέπει να αφαιρεθεί από την συνολική διαφορά της 1ης ομάδας και να προστεθεί στην συνολική διαφορά 4ης ομάδας.

Θα χρησιμοποιήσουμε ένα πίνακα Β[5] για να αποθηκεύουμε τις βαθμολογίες των ομάδων και ένα πίνακα ΣΔ[5] για να αποθηκεύουμε τις συνολικές διαφορές των ομάδων. Μία ομάδα που νικά παίρνει δύο πόντους ενώ αν χάσει παίρνει ένα πόντο.

|  |  |
| --- | --- |
| **Αλγόριθμος** Eurobasket  //Σ1[5,5], Π1[5,5], Χώρες[5]//  ! Αρχικοποίηση πινάκων Βαθμολογίας και Συνολικής Διαφοράς  **Για** i **από** 1 **μέχρι** 5  Β[i]🡨0  ΣΔ[i]🡨0  **Τέλος\_επανάληψης**  ! Υπολογισμός Βαθμολογίας και Συνολικής Διαφοράς  **Για** i **από** 1 **μέχρι** 4  **Για** j **από** i+1 **μέχρι** 5  **Αν** Σ1[i,j] = ‘N’ **τότε** ! Σε περίπτωση νίκης  Β[i] 🡨 Β[i] + 2  Β[j] 🡨 Β[j] + 1  ΣΔ[i] 🡨 ΣΔ[i] + Π1[i,j]  ΣΔ[j] 🡨 ΣΔ[j] - Π1[i,j]  **αλλιώς** ! Σε περίπτωση ήττας  Β[i] 🡨 Β[i] + 1  Β[j] 🡨 Β[j] + 2  ΣΔ[i] 🡨 ΣΔ[i] - Π1[i,j]  ΣΔ[j] 🡨 ΣΔ[j] + Π1[i,j]  **Τέλος\_αν**  **Τέλος\_επανάληψης**  **Τέλος\_επανάληψης** | !Οι πίνακες Β και ΣΔ έχουν τη βαθμολογία και την συνολική  ! διαφορά των ομάδων του ομίλου. Στη !συνέχεια του αλγορίθμου  ! μπορεί να γίνει ταξινόμηση π.χ. με τη μέθοδο της φυσαλίδας για  ! να καταταχτούν οι ομάδες σε φθίνουσα σειρά. Στην ταξινόμηση  !όμως με βάση τη βαθμολογία μπορούμε να ελέγχουμε σε  !περίπτωση ισοβαθμίας και τη διαφορά πόντων. Ακόμη η  ! ταξινόμηση  ! γίνεται κατά φθίνουσα σειρά δηλαδή στην πρώτη θέση του  ! πίνακα αναδύεται το μεγαλύτερο στοιχείο. Τέλος μαζί με τη  !βαθμολογία και τη συνολική διαφορά πόντων πρέπει να γίνει  ! αντιμετάθεση και των !ονομάτων των χωρών για να ακολουθεί  ! κάθε χώρα τη βαθμολογία της.  **Για** i **από** 2 **μέχρι** 5  **Για** j **από** 5 **μέχρι** i **με\_βήμα** -1  **Αν** Β [j-1] < Β[j] **τότε**  προσωρινή1 🡨 Β [j]  Β [j] 🡨 Β [j-1]  Β [j-1] 🡨 προσωρινή1  προσωρινή2 🡨 ΣΔ [j]  ΣΔ [j] 🡨 ΣΔ [j-1]  ΣΔ [j-1] 🡨 προσωρινή2  προσωρινή3 🡨 Χώρες [j]  Χώρες [j] 🡨 Χώρες [j-1]  Χώρες [j-1] 🡨 προσωρινή3  **αλλιώς\_αν** (Β [j-1] = Β[j]) **ΚΑΙ** (ΣΔ[j-1]<ΣΔ[j]) **τότε**  προσωρινή1 🡨 Β [j]  Β [j] 🡨 Β [j-1]  Β [j-1] 🡨 προσωρινή1  προσωρινή2 🡨 ΣΔ [j]  ΣΔ [j] 🡨 ΣΔ [j-1]  ΣΔ [j-1] 🡨 προσωρινή2  προσωρινή3 🡨 Χώρες [j]  Χώρες [j] 🡨 Χώρες [j-1]  Χώρες [j-1] 🡨 προσωρινή3  **Τέλος\_αν**  **Τέλος\_επανάληψης** Τέλος\_επανάληψης !Εκτύπωση πίνακα  **Για** i **από** 1 **μέχρι** 5  **Εμφάνισε** ‘Στην ‘, i, ‘θέση είναι η ομάδα’, Χώρες[i], ‘ με βαθμολογία’, Β[i], ‘ και διαφορά πόντων’, ΣΔ[i]  **Τέλος\_επανάληψης**  **Τέλος** Eurobasket |