

Κεφάλαιο 3 Δομές Δεδομένων και Αλγόριθμοι - Τετράδιο Μαθητή

Στην τάξη

ΔΤ1. Σε μία κατασκήνωση υπάρχουν 300 παιδιά και καθένα από αυτά έχει μοναδικό αριθμό από το 1 έως και το 300 που του αντιστοιχεί. Για κάθε παιδί είναι γνωστή η ηλικία του. Να χρησιμοποιηθεί η δομή του πίνακα για να αποθηκεύονται οι ηλικίες των παιδιών και να βρεθεί ο κατάλληλος αλγόριθμος υπολογισμού του μικρότερου και μεγαλύτερου σε ηλικία παιδιού και να εκτυπώνεται τόσο η ηλικία όσο και ο κωδικός του μικρότερου και μεγαλύτερου παιδιού

Αλγόριθμος Κατασκήνωση

Δεδομένα // ΗΛΙΚΙΑ //

ελάχιστος ← ΗΛΙΚΙΑ[1]

θέση_ελάχιστος ← 1

μέγιστος ← ΗΛΙΚΙΑ[1]

θέση_μέγιστος ← 1

Για i από 2 μέχρι 300

Αν (ΗΛΙΚΙΑ[i] < ελάχιστος) τότε

ελάχιστος ← ΗΛΙΚΙΑ[i]

θέση_ελάχιστος ← i

Τέλος_Αν

Αν (ΗΛΙΚΙΑ[i] > μέγιστος) τότε

μέγιστος ← ΗΛΙΚΙΑ[i]

θέση_μέγιστος ← i

Τέλος_Αν

Τέλος_Επανάληψης

Εκτύπωσε "Η μικρότερη ηλικία είναι ", ελάχιστος, " απο το παιδί με κωδικό", θέση_ελάχιστος

Εκτύπωσε "Η μεγαλύτερη ηλικία είναι ", μέγιστος, " απο το παιδί με κωδικό", θέση_μέγιστος

Τέλος Κατασκήνωση

ΔΤ2. Ο αλγόριθμος της φουσαλίδας όπως διατυπώθηκε στην παράγραφο 3.7 έχει το μειονέκτημα ότι δεν είναι αρκετά «έξυπνος» ώστε να διαπιστώνει στην αρχή ή στο μέσο της διαδικασίας αν ο πίνακας είναι ταξινομημένος. Να σχεδιασθεί μία παραλλαγή του αλγορίθμου αυτού που να σταματά όταν διαπιστωθεί ότι τα στοιχεία του πίνακα είναι ήδη ταξινομημένα.

Υπόδειξη: Να χρησιμοποιήσετε μία βοηθητική μεταβλητή που να ελέγχει το τέλος κάθε επανάληψης του εξωτερικού βρόχου (Για i από 2 μέχρι n) αν για την τρέχουσα τιμή του i έγιναν αντιμεταθέσεις στοιχείων

```

Αλγόριθμος Κεφ3ΔΤ2 ! Με Όσο .. επανάλαβε
  Δεδομένα // table, n //
  Sorted ← Ψευδής
  i ← 2
  Όσο (i ≤ n) και (Sorted = Ψευδής) επανάλαβε
    Sorted ← Αληθής
    Για j απο n μέχρι i με_βήμα -1
      Αν table[j - 1] > table[j] τότε
        Sorted ← Ψευδής
        Αντιμετάθεσε table[j], table[j - 1]

      Τέλος_αν
    Τέλος_επανάληψης
    i ← i + 1
  Τέλος_επανάληψης
Αποτελέσματα // table //
Τέλος Κεφ3ΔΤ2

```

```

Αλγόριθμος Κεφ3ΔΤ2 ! Με Αρχή_επανάληψης .. μέχρις_ότου
  Δεδομένα // table, n //
  i ← 2
  Αρχή_επανάληψης
    Sorted ← Αληθής

    Για j απο n μέχρι i με_βήμα -1
      Αν table[j - 1] > table[j] τότε
        Sorted ← Ψευδής
        Αντιμετάθεσε table[j - 1], table[j]

    Τέλος_αν
    Τέλος_επανάληψης
    i ← i + 1

  Μέχρις_ότου i > n ή Sorted = Αληθής
  Αποτελέσματα // table //
Τέλος Κεφ3ΔΤ2

```

ΔΤ6. Μία οικολογική οργάνωση διαθέτει στοιχεία για το ποσοστό δασών για 50 διαφορετικές χώρες. Χρειάζεται να πάρει απόφαση για να διοργανώσει μία εκδήλωση διαμαρτυρίας στις 10 χώρες που έχουν το χαμηλότερο ποσοστό δασών. Να δοθεί αλγόριθμος που θα ταξινομή τα ποσοστά δασών των χωρών με χρήση της μεθόδου της ευθείας ανταλλαγής και θα εκτυπώνει τις 10 χώρες στις οποίες θα διοργανωθούν οι εκδηλώσεις

```

Αλγόριθμος Δάση
  Δεδομένα // ΧΩΡΑ, ΠΟΣΟΣΤΟ //
  Για i από 2 μέχρι 10 ! ενδιαφέρομαι μόνο για τα 10
    Για j από 50 μέχρι i με_βήμα -1 ! αύξουσα ταξινόμηση
      Αν ΠΟΣΟΣΤΟ[j-1] > ΠΟΣΟΣΤΟ[j] τότε
        Αντιμετάθεσε ΠΟΣΟΣΤΟ[j-1], ΠΟΣΟΣΤΟ[j]
        Αντιμετάθεσε ΧΩΡΑ[j-1], ΧΩΡΑ[j]
      Τέλος_αν

```

Τέλος_επανάληψης
Τέλος_επανάληψης
Για i από 1 μέχρι 10 ! εκτύπωση των 10 πρώτων τιμών
Εκτύπωσε ΧΩΡΑ[i]
Τέλος_Επανάληψης
Τέλος Δάση

Στο σπίτι

ΔΣ1. Δίνεται ο παρακάτω πίνακας από αντιστοιχίσεις νομισμάτων διαφόρων κρατών:

Νόμισμα χώρας	Αγορά	Πώληση
Δολλάριο ΗΠΑ	1.01	1.03
Δολλάριο Καναδά	0.65	0.66
Λίρα Αγγλίας	1.57	1.59
Γεν Ιαπωνίας	0.01	0.012
Φράγκο Ελβετίας	0.67	0.68

Να γραφτεί ένας αλγόριθμος που θα κάνει μετατροπές ενός ποσού από τα ξένα νομίσματα σε ευρώ και από ευρώ στο αντίστοιχο ξένο νόμισμα

Θεωρούμε δεδομένο δισδιάστατο πίνακα ΙΣΟΤΙΜΙΑ[5, 2] που περιέχει τις τιμές της εκφώνησης. Κάθε γραμμή αντιστοιχεί σε ένα νόμισμα και κάθε στήλη σε αγορά ή πώληση. Ο αλγόριθμος θα διαβάζει το ποσό που θα μετατραπεί, τον αύξων αριθμό του νομίσματος εισόδου (αν έχει τιμή 1-5 αντίστοιχα με τον πίνακα, αν έχει τιμή 6 εννοείται το ευρώ) και τον αύξων αριθμό του νομίσματος εξόδου

Αλγόριθμος Ισοτιμία
Δεδομένα // ΙΣΟΤΙΜΙΑ //
Διάβασε νόμισμα_είσοδος, νόμισμα_έξοδος, ποσό
Αν νόμισμα_είσοδος = 6 **τότε**
 ποσό_μετατροπή ← ποσό / ΙΣΟΤΙΜΙΑ[νόμισμα_έξοδος, 2]
Αλλιώς
 ποσό_μετατροπή ← ποσό * ΙΣΟΤΙΜΙΑ[νόμισμα_είσοδος, 1]
Τέλος_αν
Εκτύπωσε ποσό_μετατροπή
Τέλος Ισοτιμία

ΔΣ2. Κατά τη διάρκεια ενός πρωταθλήματος μπάσκετ καταγράφεται ο αριθμός των πόντων που έχουν βάλει 5 παίκτες σε 5 διαφορετικά παιχνίδια. Να γραφτεί αλγόριθμος που θα σε βοηθήσει να κρατήσεις σε ένα δισδιάστατο πίνακα αυτά τα στοιχεία και στη συνέχεια να υπολογίσεις

τον παίκτη που έχει πετύχει το μεγαλύτερο αριθμό πόντων από όλα τα παιχνίδια

Θεωρούμε δεδομένο δισδιάστατο πίνακα ΠΟΝΤΟΙ[5, 5] που περιέχει τους πόντους του παίκτη σε κάθε αγώνα. Κάθε γραμμή του πίνακα αντιστοιχεί σε έναν παίκτη και κάθε στήλη σε έναν αγώνα. Θα δημιουργήσουμε ένα πίνακα με τους συνολικούς πόντους κάθε παίκτη - μονοδιάστατος 5 θέσεων και θα εντοπίσουμε σε αυτόν το μέγιστο στοιχείο και τη θέση του

Αλγόριθμος Πρωτάθλημα

Δεδομένα // ΠΟΝΤΟΙ //

Για i από 1 μέχρι 5

ΣΥΝ_ΠΟΝΤΟΙ[i] ← 0

Για j από 1 μέχρι 5

ΣΥΝ_ΠΟΝΤΟΙ[i] ← ΣΥΝ_ΠΟΝΤΟΙ[i] + ΠΟΝΤΟΙ[i, j]

Τέλος_Επανάληψης

Τέλος_Επανάληψης

μέγιστος ← ΣΥΝ_ΠΟΝΤΟΙ[1]

θέση_μέγιστος ← 1

Για i από 2 μέχρι 5

Αν (ΣΥΝ_ΠΟΝΤΟΙ[i] > μέγιστος) τότε

μέγιστος ← ΣΥΝ_ΠΟΝΤΟΙ[i]

θέση_μέγιστος ← i

Τέλος_Αν

Τέλος_Επανάληψης

Εκτύπωσε "Ο παίκτης με το μεγαλύτερο score είναι ο", θέση_μέγιστος, "με", μέγιστος, "πόντους"

Τέλος Πρωτάθλημα

ΔΣ3. Έστω ότι θέλουμε να διατάξουμε τους μαθητές μίας τάξης κατά φθίνουσα σειρά ύψους. Η τεχνική που θα ακολουθήσουμε είναι η εξής. Αρχικά, τοποθετούμε τους μαθητές σε μία τυχαία σειρά. Κατόπιν συγκρίνουμε το δεύτερο με τον πρώτο και αν χρειασθεί τους αντιμεταθέτουμε ώστε πρώτος να είναι ο ψηλότερος. Στη συνέχεια θεωρούμε τον τρίτο και τον τοποθετούμε στη σωστή σειρά σε σχέση με τον πρώτο και το δεύτερο. Κατ' αυτόν τον τρόπο συνεχίζουμε μέχρι να τοποθετήσουμε στη σωστή σειρά όλους τους μαθητές. Να σχεδιασθεί ένας αλγόριθμος που να υλοποιεί αυτή τη μέθοδο ταξινόμησης

Αλγόριθμος Κεφ3ΔΣ3

Δεδομένα // table, n //

Για i από 2 μέχρι n

Για j από i μέχρι 2 με_βήμα -1

Αν table[j] > table[j - 1] τότε

Αντιμετάθεσε table[j], table[j - 1]

Τέλος_αν

Τέλος_επανάληψης

Τέλος_επανάληψης

Εμφάνισε P[i]
Τέλος_επανάληψης
Αλλιώς
Εμφάνισε " Δεν βρέθηκε ο αριθμός ", key
Τέλος_Αν
Τέλος Αναζήτηση