**Ερώτημα 1ο** (20%)

(α) Έστω , όπου Να δείξετε ότι

(β) Να δείξετε ότι

**Απάντηση**

**(α)** Για κάθε ισχύει ότι Επομένως,

Στη συνέχεια θα δείξουμε ότι υπάρχουν αριθμοί τέτοιοι ώστε, για κάθε ισχύει

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

Επειδή έχουμε ότι για κάθε ισχύει και επομένως

Η συνάρτηση είναι αύξουσα για n>0, αφού

Η μικρότερη τιμή για την οποία είναι Επειδή η συνάρτηση είναι αύξουσα, για κάθε έχουμε

Θέτοντας στην ανισότητα (1) έχουμε

Άρα είναι δύο κατάλληλοι αριθμοί και επομένως

Από τα παραπάνω έχουμε ότι

**(β)**

Πρέπει να υπολογίσουμε την παράσταση

Επειδή και έχουμε

και επομένως

**Θέμα 2ο** (40%)

Τρεις αλγόριθμοι επιλύουν το ίδιο πρόβλημα. Ο αριθμός των στοιχειωδών πράξεων για καθένα από τους αλγόριθμους δίνεται από τις σχέσεις:

Αλγόριθμος Α

Αλγόριθμος B

Αλγόριθμος Γ

Ποιόν από τους τρεις αλγόριθμους θα επιλέγατε για την επίλυση του προβλήματος; Να αιτιολογήσετε την απάντηση σας.

**Απάντηση**

Για τους δύο πρώτους αλγόριθμους θα χρησιμοποιήσουμε το κεντρικό θεώρημα.

**Αλγ. Α**

Έχουμε και επομένως Επειδή έχουμε τη δεύτερη περίπτωση του κεντρικού θεωρήματος και επομένως

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

**Αλγ. B**

Έχουμε και επομένως Παρατηρούμε ότι υπάρχει , τέτοιο ώστε

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

αφού

(Για παράδειγμα, για η (3) ικανοποιείται.)

Επομένως

Επίσης παρατηρούμε ότι

για (π.χ. ).

Από τα παραπάνω προκύπτει ότι έχουμε την τρίτη περίπτωση του κεντρικού θεωρήματος και επομένως

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

**Αλγ. Γ**

Αρχικά θα μελετήσουμε την αναδρομική συνάρτηση χωρίς το πάτωμα, δηλαδή τη συνάρτηση

Έχουμε και επομένως Παρατηρούμε ότι εφόσον

|  |  |  |
| --- | --- | --- |
|  |  |  |

για κάποιο αφού

Επομένως

Επίσης παρατηρούμε ότι

για (π.χ. ).

Από τα παραπάνω προκύπτει ότι έχουμε την τρίτη περίπτωση του κεντρικού θεωρήματος και επομένως

|  |  |  |
| --- | --- | --- |
|  |  |  |

Εικάζουμε ότι η συνάρτηση είναι πολυπλοκότητας αντίστοιχης με την Αρχικώς θα δείξουμε με επαγωγή ότι .

Επαγωγική Βήμα

Έστω ότι κάτι τέτοιο ισχύει για Υπάρχουν δηλαδή , τέτοια ώστε για κάθε ισχύει Έχουμε

εφόσον

Επομένως,

Επιπλέον

και άρα

Συνολικά

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Από (2), (4), (5) έπεται ότι για να συγκρίνουμε τους τρεις αλγόριθμους πρέπει να κατατάξουμε ασυμπτωτικά τις συναρτήσεις

Προφανώς

|  |  |  |
| --- | --- | --- |
|  | , | (6) |

αφού

Επιπλέον

αφού (Ερώτημα 1β) και επομένως ασυμπτωτικά,

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

Από (6), (7) συνεπάγεται ότι ο Αλγόριθμος Α έχει τη χαμηλότερη πολυπλοκότητα και είναι αυτός που προτείνεται.

**Θέμα 3ο (40%)**

Στη συνέντευξη που δίνετε για προσληφθείτε στην εταιρεία MegaSoftware σας θέτουν το ακόλουθο πρόβλημα: πρέπει να περιγράψετε έναν «γρήγορο» αλγόριθμο ο οποίος να βρίσκει τον μεγαλύτερο από μία σειρά από ακεραίους οι οποίοι βρίσκονται αποθηκευμένοι στις θέσεις από *k* μέχρι *r* *(k<r)* ενός πίνακα *A.* Έχετε την εξής καταπληκτική ιδέα: θα προτείνετε έναν αλγόριθμο τύπου «ΔΙΑΙΡΕΙ και ΒΑΣΙΛΕΥΕ». Ειδικότερα, ο αλγόριθμος σας θα συγκρίνει *αναδρομικά* το μεγαλύτερο από τα αριστερότερα μισά στοιχεία του πίνακα με τον μεγαλύτερο από τα υπόλοιπα (δεξιότερα) μισά στοιχεία και θα επιστρέφει το μεγαλύτερο από τους δύο.

**(α)** Να δώσετε μία περιγραφή του παραπάνω αλγόριθμου σε μορφή αναδρομικής συνάρτησης η οποία να επιστρέφει το μεγαλύτερο αριθμό της σειράς.

**(β)** Για να αποδείξετε ότι ο αλγόριθμος σας είναι «γρήγορος», να υπολογίσετε την πολυπλοκότητα του.. Για ευκολία μπορείτε να θεωρήσετε ότι το πλήθος των ακεραίων είναι δύναμη του 2.

**(γ)** Ο εξεταστής σας έχει τη γνώμη ότι ο αλγόριθμος σας είναι χειρότερος από τον τετριμμένο αλγόριθμο επίλυσης του προβλήματος: σαρώνεται η σειρά των αριθμών αποθηκεύοντας σε κάθε βήμα το μεγαλύτερο ακέραιο που έχει βρεθεί μέχρι το βήμα αυτό.

Έχει δίκιο ο εξεταστής σας; Αιτιολογήστε την απάντηση σας και επιχειρηματολογήστε υπέρ της πρόσληψης σας.

**Απάντηση**

**(α)** Ο αλγόριθμος ακολουθεί

|  |  |
| --- | --- |
|  | max\_el (int A[],int k, int r) |
| 1 | if (k=r) then |
| 2 | return A[k]; |
| 3 | else |
| 4 | middle🡨(r-k+1)/2; |
| 5 | k1🡨k; r1🡨lb+middle-1; |
| 6 | k2🡨 lb+middle; r2🡨r; |
| 7 | max1🡨max\_el(A,k1,r1); |
| 8 | max2🡨max\_el(A,k2,r2); |
| 9 | if max1<max2 then |
| 10 | return max2; |
| 11 | end\_if |
| 12 | return max1; |
| 13 | end\_if |

**(β)**

Ο αλγόριθμος εκτελεί σταθερό αριθμό στοιχειωδών πράξεων σε κάθε βήμα και μια σύγκριση όταν *n=1 (k=r).* Επίσης σε κάθε κλήση πρέπει να επιλυθούν δύο προβλήματα μεγέθους μισού του αρχικού. Επομένως ο αριθμός των στοιχειωδών πράξεωνδίνεται από τον τύπο

Έχουμε και επομένως Επειδή για κάποιο έχουμε

όπου

Από την πρώτη περίπτωση του κεντρικού θεωρήματος έχουμε

**(γ)**

Ο τετριμμένος αλγόριθμος κωδικοποιείται ως εξής

|  |  |
| --- | --- |
|  | max\_el\_triv (int A[],int k, int r) |
| 1 | max\_el1🡨-∞; |
| 2 | for (i🡨k;i<=r,i++) |
| 3 | if (max\_el1<A[i]) then |
| 4 | max\_el1🡨A[i]; |
| 5 | end\_if; |
| 6 | end\_for; |
| 7 | return max\_el1; |

Παρατηρούμε ότι ο αλγόριθμος αυτός έχει πολυπλοκότητα και επομένως έχει ίδια ασυμπτωτική συμπεριφορά με τον αλγόριθμο που προτείνετε. Επομένως δεν είναι σωστό ότι ο αλγόριθμος σας είναι χειρότερος. Επίσης επειδή πρέπει να διαβαστεί τουλάχιστον μία φορά το κάθε στοιχείο η πολυπλοκότητα οποιουδήποτε αλγόριθμου δεν μπορεί να είναι χαμηλότερη του

Επιπλέον με τον αλγόριθμο που προτείνατε δείξατε ότι γνωρίζετε την τεχνική «ΔΙΑΙΡΕΙ ΚΑΙ ΒΑΣΙΛΕΥΕ» και αυτό θα πρέπει να μετρήσει θετικά στην πρόσληψη σας.