Αλγόριθμοι και Πολυπλοκότητα: 2ο Σύνολο Ασκήσεων

1. Να βρείτε την πολυπλοκότητα των παρακάτω αλγορίθμων αφού βρείτε πρώτα την αναδρομική εξίσωση.

Αλγόριθμος 1

```
int one(int n)
{
int sum=0;
int i,j;
if (n==0)
   return 1;
for(i=0;i<=n;i++)
      for (j=1;j<=n; j=2*j){
       time O(1)
      }
sum=one(n/2);
sum+=one(n/2);
return sum;
}
```

Αλγόριθμος 2

```
int two(int n)
{
int sum=0;
int i;
if (n==0)
   return 7;
for(i=0;i<=n-1;i++)
      sum+=two(i);
return sum;
}
```

2. Να επιλυθούν οι ακόλουθες αναδρομικές εξισώσεις. Δίνεται ότι το $T(n)$ είναι σταθερό για $n \leq 2$.

1. $T(n) = 3T(\frac{n}{2}) + n$
2. $T(n) = T(\sqrt{n}) + 1$
3. $T(n) = T(3n/4) + T(n/5) + cn$

3. Υποθέστε ότι έχουμε ένα πολύ μεγάλο πίνακα A, ο οποίος είναι αποθηκευμένος στη μνήμη κατά γραμμές. Θέλουμε να υπολογίσουμε τον ανάστροφο πίνακα του A, A^T και να τον αποθηκεύσουμε στη μνήμη επίσης κατά γραμμές. Επειδή ο A είναι μεγάλος, κάθε γραμμή του μπορεί να αποθηκευτεί σε μια σελίδα μνήμης ακριβώς. Η μεταφορά μνήμης είναι πολύ ακριβή, όμως, από τη στιγμή που κάποιος μεταφέρει μια σελίδα μνήμης, οι υπολογισμοί γίνονται χωρίς κόστος. Πόσες μεταφορές μνήμης χρειάζονται για να υπολογιστεί ο ανάστροφος πίνακας; Η μνήμη σας δεν είναι απαραίτητο να είναι μεγάλη, είναι αρκετό να έχετε στη διάθεσή σας μόνο δυο σελίδες μνήμη οποιαδήποτε στιγμή. Υπόδειξη : Διαίρει και βασίλευε.

4. Για την ταξινόμηση n αριθμών σε ένα μονοδιάστατο πίνακα, βρείτε το μικρότερο στοιχείο του A και αλλάξετε θέση με το A[1]. Στη συνέχεια βρείτε το δεύτερο μικρότερο στοιχείο του A και και αλλάξετε θέση με το A[2]. Συνεχίστε με αυτόν τον τρόπο για τα $n-1$ στοιχεία του A. Γράψτε τον ψευδοκώδικα του αλγορίθμου, ο οποίος είναι γνωστός ώς $selection\ sort$. Δώστε τους χρόνους της καλύτερης και χειρότερης περίπτωσης της $selection\ sort$ σε Θ συμβολισμό.

5. Θεωρείστε τον αλγόριθμο της γραμμικής αναζήτησης. Πόσα στοιχεία από την ακολουθία εισόδου χρειάζονται να ελεγχΘούν κατά μέσο όρο, υποθέτοντας ότι το υπό αναζήτηση στοιχείο μπορεί να είναι κάποιο από τα στοιχεία του πίνακα με την ίδια πιθανότητα; Πόσα στοιχεία χρειάζονται να ελεγχθούν το πολύ; Ποιοί είναι οι χρόνοι καλύτερης και χειρότερης περίπτωσης της γραμμικής αναζήτησης σε Θ συμβολισμό. Δικαιολογήστε τις απαντήσεις σας.

6. Η $insertion\ sort$ (ενθετική ταξονόμηση) μπορεί να εκφραστεί σαν μια αναδρομική διαδικασία με τον ακόλουθο τρόπο: Για να ταξινομήσουμε τη συστοιχία $A[1...n]$, ταξινομούμε αναδρομικά την υποσυστοιχία $A[1...n-1]$ και στη συνέχεια ενθέτουμε το στοιχείο $A[n]$ στην ταξινομημένη υποσυστοιχία $A[1...n-1]$. Διατυπώστε μια αναδρομική σχέση για το χρόνο εκτέλεσης αυτής της αναδρομικής εκδοχής της ενθετικής ταξονόμησης.

7. Παρατηρήστε ότι η ενθετική ταξινόμηση χρησιμοποιεί γραμμική αναζήτηση για να διατρέξει την ταξινομημένη υποσυστοιχία $A[1...j-1]$. Αν υποτεθεί ότι το κόστος προσπέλασης της μνήμης κατά την ανταλλαγή των τιμών δυο στοιχείων είναι αμελητέο, να προτείνετε μια τροποποίησή της έτσι ώστε να μειώνεται η πολυπλοκότητά της.

8. Θεωρούμε μια ταξινομημένη ακολουθία A, μεγέθους n. Να γράψετε τον ψευδοκώδικα της Δυαδικής αναζήτησης ενός στοιχείου στην A. Να αποδείξετε ότι η πολυπλοκότητα της Δυαδικής αναζήτησης είναι $\Theta(\log n)$

9. Ορθότητα της φυσαλιδωτής ταξινόμησης
Η φυσαλιδωτή ταξίνόμηση είναι ενας αρκετά διαδεδομένος αλγόριθμος ταξινόμησης. Λειτουργεί εναλλάσσοντας κατ' επανάληψη παρακείμενα στοιχεία που είναι διατεταγμένα με λάθος σειρά.

ΦΥΣΑΛΙΔΩΤΗ ΤΑΞΙΝΟΜΗΣΗ(A)
1.**για** $i \leftarrow$ έως μήκος$[A]$
2. **για** $j \leftarrow$ μήκος$[A]$ **αντίστροφα- έως** $i+1$
3. **αν** $A[j] < A[j-1]$
4. **τότε** εναλλαγή $A[j] \leftrightarrow A[j-1]$

α. Έστω ότι συμβολιζουμε την έξοδο της φυσαλιδωτής ταξινόμησης (A) με A′. Για να αποδείξουμε ότι ο αλγόριθμος είναι ορθός, αρκεί να δείξουμε ότι τερματίζει και ότι:

$$A'[1] \leq A'[2] \leq ... \leq A'[n] \tag{1}$$

όπου n =μήκος[A].

β. Διατυπώστε με ακρίβεια μια αναλλοίωτη συνθήκη για το βρόχο **για** στις γραμμές 2-4, και αποδείξτε ότι ισχύει. Η απόδειξή σας θα πρέπει να βασίζεται στη δομή της απόδειξης αναλλοίωτων συνθηκών.

γ. Χρησιμοποιώντας την "επιβεβαίωση αποτελέσματος" της αναλλοίωτης συνθήκης την οποία αποδείξατε στο υποερώτημα (β), να διατυπώσετε μια αναλλοίωτη συνθήκη για τον βρόχο **για** στις γραμμές 1-4, η οποία θα σας επιτρέψει να αποδείξετε την ανισότητα (1). Η απόδειξή σας θα πρέπει να βασίζεται στη δομή της απόδειξης των αναλλοίωτων συνθηκών.

δ. Ποιός είναι ο χρόνος εκτέλεσης χειρότερης περίπτωσης της φυσαλιδωτής ταξινόμησης; Συγκρίνετέ τον με τον αντίστοιχο χρόνο της ενθετικής ταξινόμησης.

10. Να τροποποιηθεί ο αλγόριθμος MinMax έτσι ώστε ο πίνακας $A[1..n]$ να διαιρείται σε τρία τμήματα αντί για δύο.

α. Να δοθεί ο νέος αναδρομικός αλγόριθμος.

β. Να βρεθεί η πολυπλοκότητά του.

γ. Να αποδειχθεί η ορθότητά του.

δ. Σχολιάστε τα συμπεράσματά σας.

11. Να τροποποιηθεί ο αλγόριθμος MinMax έτσι ώστε ο πίνακας $A[1..n]$ να διαιρείται σε δύο τμήματα. Το αριστερό τμήμα να περιέχει τα πρώτα δύο στοιχεία και το δεξί τμήμα περιέχει τα υπόλοιπα.

α. Να δοθεί ο νέος αναδρομικός αλγόριθμος.

β. Να βρεθεί η πολυπλοκότητά του.

γ. Να αποδειχθεί η ορθότητά του.

δ. Να συγκρίνετε τον νέο αλγόριθμο με τον αλγόριθμο που παρουσιάστηκε στην παράδοση (βλ. Διαφάνειες και Σημειώσεις).

12. Δίνεται ο n x n πίνακας $A = (a_{ij})$. Ο υπολογισμός της $||A||_1$ δίνεται από τον ακόλουθο τύπο

$$||A||_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}|$$

Για μεγάλες τιμές του n ο ανωτέρω υπολογισμός είναι πολύ αργός.

α. Ποιά η πολυπλοκότητα του υπολογισμού της $||A||_1$ με βάση τον ανωτέρω τύπο;

β. Να δοθεί αναδρομικός αλγόριθμος για τον υπολογισμό της $||A||_1$.

γ. Να βρεθεί η πολυπλοκότητά του.

δ. Να αποδειχθεί η ορθότητά του.

ε. Είναι ο νέος αλγόριθμος βέλτιστος; Δικαιολογήστε.

13. Μια ομάδα Φυσικών θέλουν να μελετήσουν τη συνολική δύναμη που ασκείται σε κάθε σωματίδιο $j = 1, \ldots, n$, που έχει φορτίο q_j, μετρώντας τη και μετά συγκρίνοντάς τη με μια υπολογιστική πρόβλεψη. Η συνολική δύναμη που ασκείται σε ένα σωματίδιο j, από τον νόμο του Coulomb, είναι ίση με

$$F_j = \sum_{i<j} \frac{Cq_iq_j}{(j-i)^2} - \sum_{i>j} \frac{Cq_iq_j}{(j-i)^2}, \quad j = 1, \ldots, n$$

Οι Φυσικοί έχουν υπολογίσει την συνολική δύναμη για κάθε σωματίδιο με βάση τον ανωτέρω τύπο, αλλά για μεγάλο πλήθος σωματιδίων είναι υπερβολικά αργός. Θέλουν την βοήθειά σας προκειμένου να μειώσουν τον χρόνο εκτέλεσης των υπολογισμών τους. Τι έχετε να τους προτείνετε; Να δικαιολογήσετε αναλυτικά τις προτάσεις σας. Πιο συγκεκριμένα

α. Να βρεθεί η πολυπλοκότητα των υπολογισμών των Φυσικών.

β. Να δοθεί αναδρομικός αλγόριθμος για τον υπολογισμό της $F_j, \quad j = 1, \ldots, n$.

γ. Να βρεθεί η πολυπλοκότητά του.

δ. Να αποδειχθεί η ορθότητά του.

ε. Είναι ο αλγόριθμός σας βέλτιστος; Δικαιολογήστε.