

Αλγόριθμοι και Πολυπλοκότητα: 2ο Σύνολο Ασκήσεων

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

Αλγόριθμος 1 Stooage Sort (A, l, r)

Require: Πίνακας n στοιχείων A και δείκτες l, r

Ensure: Ταξινομημένη έκδοση του A από το δείκτη l έως το δείκτη r

```
1: if  $A[l] > A[r]$  then
2:   Swap( $A[l], A[r]$ )
3: end if
4: if  $l + 1 \geq r$  then
5:   return
6: end if
7:  $k := \lfloor (r - l + 1) / 3 \rfloor$ 
8: Stooage Sort ( $A, l, r - k$ )
9: Stooage Sort ( $A, l + k, r$ )
10: Stooage Sort ( $A, l, r - k$ )
```

Αλγόριθμος 2

```
int one(int n)
{
int sum=0;
int i, j;
if (n==0)
  return 1;
for (i=0; i<=n; i++)
  for (j=0; j<=log(n); j++){
    loop body
  }
sum=one(n/2);
sum+=one(n/2);
return sum;
}
```

2. Έχετε να επιλέξετε ανάμεσα στους παρακάτω τρεις αλγόριθμους που επιλύουν το πρόβλημα P διάστασης n .

1. Τον αλγόριθμο A, που διαιρεί το αρχικό πρόβλημα σε 5 υποπροβλήματα διάστασης $\frac{n}{2}$, επιλύει αναδρομικά κάθε υποπρόβλημα και συνδυάζει τις λύσεις σε χρόνο $\Theta(n \log n)$.
2. Τον αλγόριθμο B, που επιλύει αναδρομικά 3 υποπροβλήματα διάστασης $n - 1$ και συνδυάζει τις λύσεις σε χρόνο $\Theta(1)$.

3. Τον αλγόριθμο C , που διαιρεί το αρχικό πρόβλημα σε 4 υποπροβλήματα διάστασης $\frac{n}{2}$, επιλύει αναδρομικά κάθε υποπρόβλημα και συνδυάζει τις λύσεις σε χρόνο $\Theta(n^2)$.

α) Ποιά είναι η πολυπλοκότητα $T(n)$ καθενός από τους παραπάνω αλγόριθμους;

β) Ποιόν θα επιλέγατε;

Δίνονται: $\log_2 5 \approx 2.32$ και $T(0) = T(1) = \Theta(1)$.

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

1. $T(n) = 3T\left(\frac{n}{2}\right) + n$

2. $T(n) = T(\sqrt{n}) + 1$

3. $T(n) = 2T(\sqrt{n}) + 1$

4. $T(n) = T(n - 1) + n$

4. Εφαρμόστε τον αλγόριθμο Merge sort για να ταξινομήσετε τον πίνακα $\{a, b, c, 16, 7, 21, 35, 12, 20\}$, όπου a, b, c τα τελευταία τρία ψηφία του αριθμού μητρώου σας. Δείξτε σχηματικά τα αναδρομικά βήματα που γίνονται κατά την εκτέλεση του αλγορίθμου. (Να δώσετε το δέντρο αναδρομικών κλήσεων).

5. Θεωρήστε τον αλγόριθμο Quick Sort για την ταξινόμηση μιας ακολουθίας a_1, a_2, \dots, a_n με $a_i \neq a_j \forall i, j$.

1. Αν τα στοιχεία $a_i, 1 \leq i \leq n$ είναι ταξινομημένα κατά αύξουσα σειρά σχεδιάστε το δέντρο των αναδρομικών κλήσεων του αλγορίθμου.

2. Να βρεθεί η πολυπλοκότητα του αλγορίθμου στη χειρίστη περίπτωση. Συγκρίνετε τη πολυπλοκότητα με αυτήν του Heap Sort στη χειρίστη περίπτωση.

3. Είναι βέλτιστος ως προς τη πολυπλοκότητα στη χειρίστη περίπτωση ο αλγόριθμος;

4. Να ταξινομήσετε κατά αύξουσα σειρά τους χαρακτήρες E, R, B, E . Να δοθεί το δέντρο των αναδρομικών κλήσεων.

6.

1. Σε ένα σωρό με n στοιχεία να αποδείξετε ότι υπάρχουν το πολύ $\lceil n/2^{h+1} \rceil$ κόμβοι ύψους h .

2. Ποιό είναι το μέγιστο και το ελάχιστο πλήθος στοιχείων σε σωρό ύψους h ;

3. Ένας πίνακας σε φθίνουσα διάταξη είναι σωρός;

4. Να ταξινομηθούν με τον αλγόριθμο Heapsort τα παρακάτω στοιχεία: 6,9,7,4,5,8.