ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ «ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ» - 25/6/2007 – ***ΣΕΙΡΑ Α***

ΕΠΙΘΕΤΟ: . . . . . ΟΝΟΜΑ: . . . . . ΕΡΓΑΣΙΑ: 6 προβλήματα ☐ (10 μον.)

ΕΞΑΜΗΝΟ: . . . . . ΑΕΜ: . . . . . 3 προβλήματα ☐ (5 μον.)

ΔΙΔΑΣΚΟΝΤΕΣ: Ι. Μανωλόπουλος - Π. Κατσαρός 2 προβλήματα ☐ (2 bonus)

ΒΑΘΜΟΣ: 1 πρόβλημα ☐ (1 bonus)

1. Θεωρήστε τη διαδικασία `Merge` που χρησιμοποιείται στην υλοποίηση του αλγορίθμου `MergeSort`:

procedure `Merge(L[1:n], low, x, high)`

Input: `L[1:n]` (μία λίστα στοιχείων με μέγεθος *n*)
`low,x,high` (δείκτες του πίνακα L[1:n]
οι υποπίνακες L[low:x], L[x+1:high] υποθέτουμε ότι είναι ταξινομημένοι σε μη φθίνουσα)
Output: ο υποπίνακας `L[low:high]` ταξινομημένος σε μη φθίνουσα σειρά

```
CurPos1 := low
CurPos2 := x+1
Counter := low
while CurPos1 ≤ x and CurPos2 ≤ high do
 if L[CurPos1] ≤ L[CurPos2] then
       Temp[Counter] := L[CurPos1]
       CurPos1 := CurPos1 + 1
 else
       Temp[Counter] := L[CurPos2]
       CurPos2 := CurPos2 + 1
 endif
 Counter := Counter + 1
endwhile
if CurPos1 > x then
 for k := CurPos2 to high do
       Temp[Counter] := L[k]
       Counter := Counter + 1
 endfor
else
 for k := CurPos1 to low do
       Temp[Counter] := L[k]
       Counter := Counter + 1
 endfor
endif
for k := low to high do
 L[k] := Temp[k]
endfor

end Merge
```

Θεωρήστε την παραλλαγή `TriMergeSort` της `MergeSort` κατά την οποία η λίστα στοιχείων υποδιαιρείται σε τρία ίσα μέρη αντί για δύο. Για διευκόλυνση της ανάλυσης μπορείτε να υποθέσετε ότι $n = 3^k$ για κάποιο μη αρνητικό ακέραιο k.

procedure `TriMergeSort(Low, High)` **recursive**

Input: `Low, High` (δείκτες ενός καθολικά προσβάσιμου πίνακα L[1:*n*] με αρχικές τιμές Low=1 και High = $n = 3^k$)
Output: ο πίνακας `L[Low:High]` ταξινομημένος σε μη φθίνουσα σειρά

```
if High ≤ Low then return endif
Third := ⌊(Low + High)/3⌋
TwoThirds := 2 * Third
call TriMergeSort(Low,Third)
```

```
call TriMergeSort(Third+1,TwoThirds)
call TriMergeSort(TwoThirds+1, High)

call Merge(Low,Third,TwoThirds)
call Merge(Low,TwoThirds,High)

end TriMergeSort
```

ΑΝΑΛΥΣΗ ΧΕΙΡΟΤΕΡΗΣ ΠΕΡΙΠΤΩΣΗΣ:

Α. Διατυπώστε μία αναδρομική εξίσωση για την πολυπλοκότητα χειρότερης περίπτωσης W(n) της διαδικασίας `TriMergeSort` για μία λίστα εισόδου μεγέθους *n* (σχολιάστε την προσέγγισή σας). **ΜΟΝΑΔΕΣ**: 1.5

Β. Επιλύστε την αναδρομική εξίσωση που διατυπώσατε για να πάρετε την πολυπλοκότητα χειρότερης περίπτωσης W(n) της διαδικασίας `TriMergeSort`. **ΜΟΝΑΔΕΣ**: 1.5

Γ. Ποιος από τους αλγορίθμους `MergeSort` και `TriMergeSort` είναι πιο αποδοτικός στη χειρότερη περίπτωση; Αιτιολογείστε την απάντησή σας. **ΜΟΝΑΔΕΣ**: 1.0

2. Έστω $f(n)=n^3/2$ και $g(n)=37n^2+120n+17$. Αποδείξτε ότι $g \in O(f)$ και επίσης ότι $f \notin O(g)$. **ΜΟΝΑΔΕΣ**: 2.0

3. Περιγράψτε τα βήματα υπολογισμού του πηλίκου δύο αριθμών σε δυαδική μορφή. Αν υποθέσουμε ότι ο πρώτος αριθμός έχει k_1 ψηφία και ο δεύτερος έχει k_2 ψηφία ποια είναι η πολυπλοκότητα της πράξης της διαίρεσης;

ΜΟΝΑΔΕΣ: 2.0

4. Να αποδείξετε ότι:

$$2n^2 2^n + n \log n = \Theta(n^2 2^n)$$

ΜΟΝΑΔΕΣ: 2.0