ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΞΕΤΑΣΕΙΣ ΣΤΟ ΜΑΘΗΜΑ «ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ» - 16/1/2006

ΕΠΙΘΕΤΟ: . . . . ΟΝΟΜΑ: . . . .
ΕΞΑΜΗΝΟ: . . . . ΑΕΜ: . . . .
ΔΙΔΑΣΚΟΝΤΕΣ: Ι. Μανωλόπουλος - Π. Κατσαρός

1. Να επιλύσετε την αναδρομή

$$T(n)=T(n-1)+T(n-2),\ n \geq 2$$

με T(0)=0, T(1)=1, που αντιστοιχεί στον αλγόριθμο υπολογισμού των όρων της σειράς Fibonacci:

```
function fib(n)
        if n < 2 then return n
        else return fib(n-1) + fib(n-2)
```

Ποια είναι η πολυπλοκότητα του αλγορίθμου; Σχολιάστε το ενδεχόμενο βελτίωσης του παραπάνω αλγορίθμου.

ΜΟΝΑΔΕΣ 2.0

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. ΤΑΞΙΝΟΜΗΣΗ **BINGO**

Ο αλγόριθμος αυτός δίνει λύση στο πρόβλημα της ταξινόμησης λίστας με *n* επαναλαμβανόμενα στοιχεία. Έτσι, το μέγεθος εισόδου BingoSort(*n*, *m*) ορίζεται ως συνάρτηση του αριθμού *n* των στοιχείων της λίστας και του αριθμού *m* των διακριτών στοιχείων που αυτή περιέχει, όπου το *m* θεωρούμε ότι είναι κατά πολύ μικρότερο του *n* ($m <<< n$).

```
function BingoSort(L[1:n])
// Πρώτα βρίσκουμε στη λίστα L το ελάχιστο και το μέγιστο στοιχείο, με τη συνάρτηση MaxMin
MaxMin(L[1:n], MaxValue, MinValue)
Bingo ← MinValue
NextAvail ← 1
NextBingo ← MaxValue
while Bingo < MaxValue do
        StartPos ← NextAvail
        for i ← StartPos to n do
                if L[i] = Bingo then
                        interchange (L[i], L[NextAvail])
                        NextAvail ← NextAvail + 1
                else
                        if L[i] < NextBingo then NextBingo ← L[i] endif
                endif
        endfor
        Bingo ← NextBingo
        NextBingo ← MaxValue
endwhile
end BingoSort
```

Μπορεί να αποδειχθεί ότι αν η λίστα περιέχει επαναλαμβανόμενες εμφανίσεις *m* διακριτών στοιχείων, τότε ο αλγόριθμος τερματίζει μετά από *m*-1 επαναλήψεις, έχοντας ολοκληρώσει την τοποθέτηση όλων των στοιχείων με τιμή μικρότερη του μέγιστου, στη σωστή τους θέση. Στις τελευταίες θέσεις της λίστας απομένουν όλα τα στοιχεία που είναι ίσα με το μέγιστο.

Α. Εφαρμόστε την ταξινόμηση BingoSort στη λίστα με στοιχεία τα 23, 10, 15, 10, 10, 23, 15, 23, 23, 10 (ΜΟΝΑΔΑ 1.0)

Β. Με ποια είσοδο BingoSort(n, m) επιτυγχάνεται η καλύτερη περίπτωση εκτέλεσης και με ποια είσοδο BingoSort(n, m) επιτυγχάνεται η χειρότερη περίπτωση εκτέλεσης; ΜΟΝΑΔΑ: 1.0

Γ. Υπολογίστε την πολυπλοκότητα του αλγορίθμου στην καλύτερη περίπτωση και στη χειρότερη περίπτωση. Πότε συμφέρει η χρήση του BingoSort σε σχέση με τον αλγόριθμο MergeSort; ΜΟΝΑΔΑ: 2.0

4. Να υπολογιστεί η πολυπλοκότητα αλγορίθμου που το κόστος εκτέλεσής του προσδιορίζεται από την αναδρομική εξίσωση

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + n$$

ΜΟΝΑΔΕΣ 2.0