



Θ.Ε. ΠΛΣ50 (2007-2008) – ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ Ε10

Στόχος

Η εργασία αυτή στοχεύει να προετοιμάσει τους φοιτητές για τις τελικές εξετάσεις. Ενθαρρύνεστε να ασχοληθείτε με την εργασία χρησιμοποιώντας χρονόμετρο.

ΘΕΜΑ 1

Ποιές είναι οι τιμές των *trouble* και **pt* μετά την εκτέλεση του παρακάτω αποσπάσματος προγράμματος;

```
double trouble = 13.13;  
double *pt;  
pt = &trouble;  
*pt = 9.9;
```

Ποιά από τις παρακάτω προτάσεις είναι σωστή;

- a) Η *trouble* είναι ίση με 13.13 και η **pt* είναι ίση με 9.9.
- b) Η *trouble* είναι ίση με 9.9 και η **pt* είναι ίση με 13.13.
- c) Η *trouble* είναι απροσδιόριστη και η **pt* είναι ίση με 9.9.
- d) Η *trouble* είναι ίση με 9.9 και η **pt* είναι ίση με 9.9.

| | | | |
|----------------------------|----------------------------|----------------------------|----------------------------|
| <input type="checkbox"/> a | <input type="checkbox"/> b | <input type="checkbox"/> c | <input type="checkbox"/> d |
|----------------------------|----------------------------|----------------------------|----------------------------|

ΘΕΜΑ 2

Ποιά από τις παρακάτω προτάσεις είναι σωστή;

- a) Οι κλήσεις συναρτήσεων γίνονται πάντα με call-by-value
- b) Όλα τα ορίσματα μίας συνάρτησης πρέπει να έχουν τον ίδιο τύπο
- c) Στο πέρασμα πίνακα ως παραμέτρου μίας συνάρτησης εμπεριέχεται χειρισμός δείκτη
- d) Κάθε συνάρτηση πρέπει να επιστρέφει τιμή.

| | | | |
|----------------------------|----------------------------|----------------------------|----------------------------|
| <input type="checkbox"/> a | <input type="checkbox"/> b | <input type="checkbox"/> c | <input type="checkbox"/> d |
|----------------------------|----------------------------|----------------------------|----------------------------|

ΘΕΜΑ 3

Μια συνάρτηση έχει τον ακόλουθο ορισμό prototype.

```
void example (int x, int y, float*z);
```

Ποιά/ές από τις ακόλουθες κλήσεις είναι νόμιμη/ες;

| | | | |
|--|--|--|--|
| <input type="checkbox"/> example(7,2,y); | <input type="checkbox"/> example(z,z,3); | <input type="checkbox"/> example(z,z,x); | <input type="checkbox"/> example(7,2); |
|--|--|--|--|

ΘΕΜΑ 4

Ποιες από τις ακόλουθες εντολές switch εμφανίζουν τις σωστές μέρες της εβδομάδας, όπου 0=Sunday, 1=Monday and 2=Tuesday, ενώ οι άλλες μέρες αγνοούνται;



| | | |
|--|--|---|
| <pre>Switch (day) { case(0): printf("Sun"); break; case(1); printf("Mon"); break; default: printf("Tue"); break; }</pre> | <pre>switch (day) { case(0): printf("Sun"); case(1); printf("Mon"); case(2): printf("Tue"); break; }</pre> | <pre>switch (day) { case(0): printf("Sun"); break; case(1); printf("Mon"); break; case(2): printf("Tue"); }</pre> |
|--|--|---|

ΘΕΜΑ 5

Ποια από τις παρακάτω εντολές θέτει τη μεταβλητή *i* ίση με το μήκος της συμβολοσειράς *a*;

- A. `i = strlen(a);`
B. `i = 0; while (a[i] != '\0') ++i;`
Γ. `i = 0; while (a[i] != '\0') i++;`
Δ. `for (i = 0, b = a; *b != '\0'; b++) i++;`
Ε. `for (i = 0; a[i] != '\0'; i++) ;`
ΣΤ. `b = a; while (*b++) ; i = b-a-1;`
Ζ. `b = a; while (*b) b++; i = b-a;`

☐ A, B και Γ

☐ A, B, Δ και ΣΤ

☐ A, B, Ε και Ζ

☐ A, B, Γ, Δ και ΣΤ

☐ Όλες

ΘΕΜΑ 6

Τι θα εμφανιστεί από την εκτέλεση του παρακάτω κώδικα στην οθόνη;

```
int cnt = 0;
for (i = 0; i < 10; i++)
    for (j = 0; j < 10; j++)
        for (k = 0; k < 10; k++)
            if (2*i + j >= 3*k)
                cnt++;
printf("%d\n", cnt);
```

☐ 0

☐ 100

☐ 101

☐ 517

☐ 1000

ΘΕΜΑ 7

Θεωρείστε ότι στην παρακάτω συνάρτηση δίνεται ως είσοδος μία λίστα της οποίας το περιεχόμενο είναι ίσο με (9 8 1 12 8 2 5 7).

(Η κορυφή της λίστας περιέχει την τιμή 9. Ο next σύνδεσμος του τελευταίου στοιχείου είναι ίσος με NULL.) Ποια τιμή θα επιστρέψει η συνάρτηση;

```
typedef struct node *Link;
struct Node {int key; Link next; };
Int f(Link list)
{
    int a,b;
    if (list == NULL)
```



```
        return 0;
a = list->key;
b = f(list->next);
if (a > b)
    return a;
else
    return b;
}
```

☐ 8☐ 7☐ 1☐ 12

ΘΕΜΑ 8

Ποιο είναι το αποτέλεσμα της εκτέλεσης των παρακάτω εντολών;

```
y = z; z = w; w = y;
```

A. Ανταλλάσσει τις τιμές των μεταβλητών y και w.

B. Ανταλλάσσει τις τιμές των μεταβλητών y και z.

Γ. Ανταλλάσσει τις τιμές των μεταβλητών w και z.

Δ. Τίποτα από τα παραπάνω

☐ A☐ B☐ Γ☐ Δ

ΘΕΜΑ 9

Θεωρείστε το παρακάτω πρόγραμμα που χρησιμοποιεί μια FIFO ουρά. Τι θα εμφανιστεί στην οθόνη μετά την ολοκλήρωση της εκτέλεσης του παρακάτω προγράμματος;

Η συνάρτηση QUEUEinit() αρχικοποιεί την ουρά, η συνάρτηση QUEUEput() εισάγει ένα στοιχείο στην ουρά, ενώ η συνάρτηση QUEUEget() διαγράφει ένα στοιχείο από την ουρά και το επιστρέφει στο κυρίως πρόγραμμα.

```
#include <stdio.h>
#include "QUEUE.h"
int main(void) {
    int i;

    QUEUEinit();
    for (i = 0; i < 10; i++)
        QUEUEput(i);

    while (!QUEUEisempty()) {
        for (i = 0; i < 3; i++)
            QUEUEput(QUEUEget());
        printf("%d ", QUEUEget());
    }
    printf("\n");
    return 0;
}
```

A. 0 1 2 3 4 5 6 7 8 9

B. 1 3 5 7 9 2 6 0 8 4



Γ. 3 7 1 6 2 9 8 0 5 4

Δ. 4 5 0 8 9 2 6 1 7 3

Ε. 4 5 6 7 8 9 0 1 2 3

ΣΤ. 4 9 5 1 8 7 0 3 6 2

Ζ. 9 8 7 6 5 4 3 2 1 0

Η. Θα οδηγηθούμε σε λάθος λόγω προσπάθειας προσπέλασης ενός πίνακα εκτός των ορίων του.

Θ. Θα οδηγηθούμε σε queue underflow λάθος.

Ι. Τίποτα από τα παραπάνω.

| | | | | | | | | | |
|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|-----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| <input type="checkbox"/> Α | <input type="checkbox"/> Β | <input type="checkbox"/> Γ | <input type="checkbox"/> Δ | <input type="checkbox"/> Ε | <input type="checkbox"/> ΣΤ | <input type="checkbox"/> Ζ | <input type="checkbox"/> Η | <input type="checkbox"/> Θ | <input type="checkbox"/> Ι |
|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|-----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|

ΘΕΜΑ 10

Ο παρακάτω κώδικας υλοποιεί μια διπλά συνδεδεμένη λίστα, με τις καθολικές μεταβλητές Front και Rear να δείχνουν στην αρχή και στο τέλος της λίστας. Οι τέσσερις συναρτήσεις insert_front, insert_rear, delete_front και delete_rear, των οποίων τα σώματα παραλείπονται, υλοποιούν την προσθήκη ενός κόμβου στην αρχή και στο τέλος της λίστας, και τη διαγραφή ενός κόμβου από την αρχή και το τέλος της λίστας αντίστοιχα. Αντιστοιχείστε τις τέσσερις συναρτήσεις με τα τμήματα κώδικα που ακολουθούν.

```
struct double_queue {
    int data;
    struct double_queue* previous;
    struct double_queue* next;
};

typedef struct double_queue* pt;

pt Front=NULL;
pt Rear=NULL;

void insert_front(int key)
{
}

void insert_rear(int key)
{
}

void delete_front()
{
}

void delete_rear()
{
}
```

| | |
|----------|---|
| A | <pre>pt node=(pt) malloc(sizeof(struct double_queue)); node->previous=Rear; Rear=node;</pre> |
|----------|---|



| | |
|----------|---|
| | <pre>node->next=NULL; node->data=key; if (node->previous==NULL) Front=Rear; else node->previous->next=node;</pre> |
| Β | <pre>if (Rear!=NULL) { pt node=Rear; Rear=Rear->previous; free(node); if (Rear==NULL) Front=NULL; else Rear->next=NULL; }</pre> |
| Γ | <pre>pt node=(pt) malloc(sizeof(struct double_queue)); node->next=Front; Front=node; node->previous=NULL; node->data=key; if (node->next==NULL) Rear=Front; else node->next->previous=node;</pre> |
| Δ | <pre>if (Front!=NULL) { pt node=Front; Front=Front->next; free(node); if (Front==NULL) Rear=NULL; else Front->previous=NULL; }</pre> |

| | | | |
|---------------------------------------|--------------------------------------|---------------------------------------|--------------------------------------|
| <input type="checkbox"/> insert_front | <input type="checkbox"/> insert_rear | <input type="checkbox"/> delete_front | <input type="checkbox"/> delete_rear |
|---------------------------------------|--------------------------------------|---------------------------------------|--------------------------------------|

ΘΕΜΑ 11

Ποια θα είναι η τιμή της μεταβλητής M μετά την εκτέλεση του παρακάτω τμήματος κώδικα;

```
int A[4][4]={ {7,8,2,9}, {2,4,3,1}, {5,2,3,6}, {2,1,5,0} };

int i, j;
int M, M0;
M=1000;
for(i=0;i<4;i++) {
    M0=-1000;
    for(j=0;j<4;j++)
        if (A[i][j]>M0)
            M0=A[i][j];
    if (M0<M)
        M=M0;
```



}

☐ 4

☐ 5

☐ 6

☐ 9

ΘΕΜΑ 12

Ποια θα είναι η τιμή της μεταβλητής y μετά την εκτέλεση του παρακάτω κώδικα;

```
int x[] = { 1, 4, 8, 5, 1, 4 };  
int *ptr, y;  
ptr = x + 4;  
y = ptr - x;
```

☐ -3

☐ 0

☐ 4

☐ $4 + \text{sizeof}(\text{int})$

☐ $4 * \text{sizeof}(\text{int})$

ΘΕΜΑ 13

Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω εντολών;

```
int main()  
{  
    int x=0;  
    for(;x<20 && printf("%d ",x);x++)  
        switch(x)  
        {  
            case 0:x++,x*=2;  
            case 1:x+=2;  
            case 99:x+=6;  
            default:x+=3;  
            break;  
        }  
}
```

☐ 0 12 16

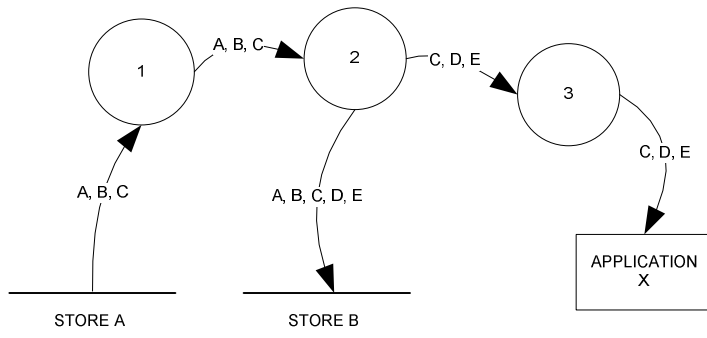
☐ 0 13 17

☐ 0 14 18

☐ 0 11 15 18

ΘΕΜΑ 14

Θεωρήστε το ακόλουθο ΔΡΔ:



Έστω οι παρακάτω προτάσεις:



Π1. Ο MM 1 δεν χρειάζεται

Π3. Ο MM 1 δεν είναι ΚΜ

Π5. Ο MM 2 προσφέρεται για ΚΜ

Π7. Τα "STORE A", "STORE B" μπορούν να συγχωνευθούν

Ποιές από αυτές είναι ΣΩΣΤΕΣ;

Α. Όλες

Γ. Οι Π3, Π4, Π5, Π6

Π2. Ο MM 3 δεν χρειάζεται

Π4. Το "APPLICATION X" είναι εξωτερική εφαρμογή λογισμικού

Π6. Το "APPLICATION X" θα μπορούσε να διαβάσει τα δεδομένα C, D, E κατευθείαν από το "STORE B".

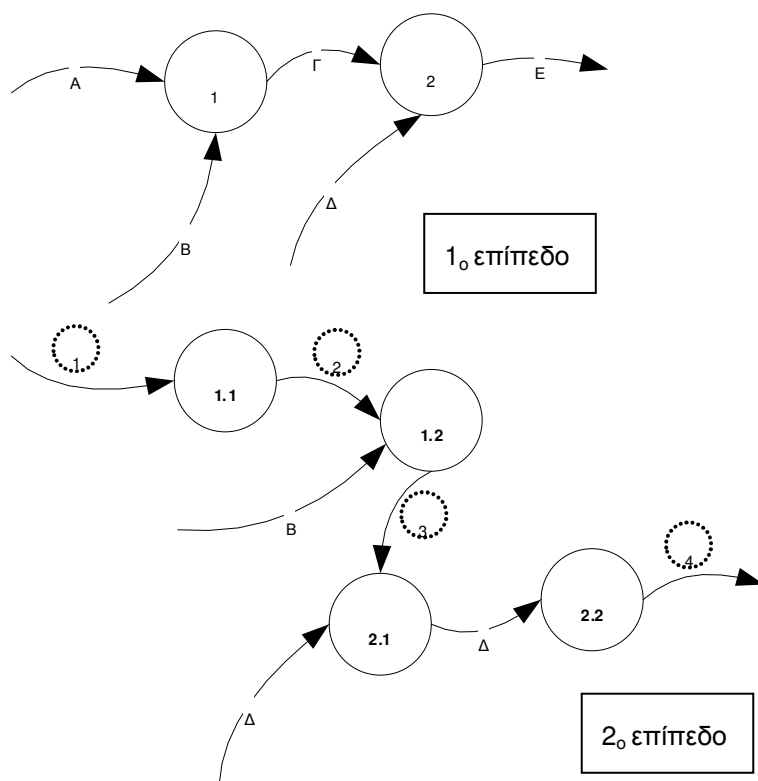
Π8. Δεν υπάρχει αλληλεπίδραση με πληκτρολόγιο και οθόνη

Β. Οι Π1, Π2, Π7

Δ. Οι Π3, Π4, Π5, Π8

ΘΕΜΑ 15

Έστω τα ακόλουθα ΔΡΔ επιπέδου 1 και 2 αντίστοιχα. Στο επίπεδο 2 ο μετασχηματισμός 1 αναλύεται στους μετασχηματισμούς 1.1 και 1.2 και ο μετασχηματισμός 2 στους μετασχηματισμούς 2.1 και 2.2. Συμπληρώστε τα δεδομένα που δεν κατονομάζονται.



| | | | |
|---|---|---|---|
| <input type="checkbox"/> 1:A, 2:Γ, 3:Γ, 4:Δ | <input type="checkbox"/> 1:A, 2:Γ, 3:Γ, 4:Ε | <input type="checkbox"/> 1:A, 2:X, 3:Γ, 4:Ε | <input type="checkbox"/> 1:Γ, 2:A, 3:B, 4:Δ |
|---|---|---|---|

ΘΕΜΑ 16

Δύο δυαδικά δέντρα είναι ισομορφικά αν υπάρχει 1-1 αντιστοιχία ανάμεσα στους κόμβους του ενός δέντρου με τους κόμβους του άλλου δέντρου και αν η ύπαρξη ακμής ανάμεσα σε δύο κόμβους στο ένα

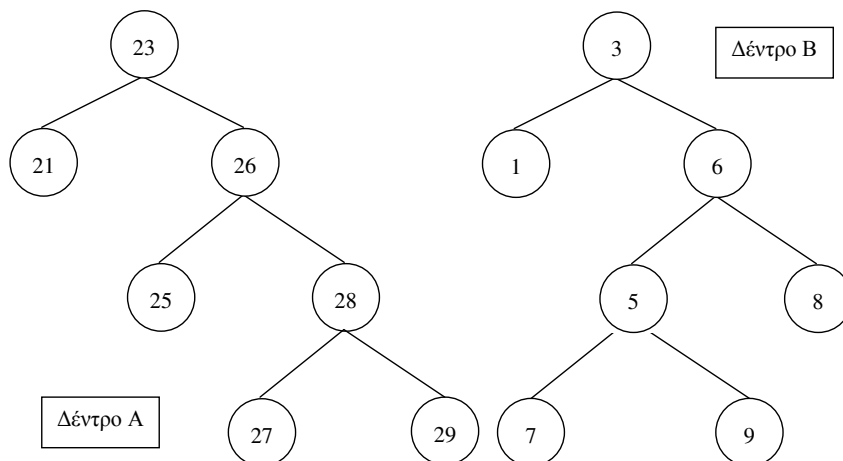


δέντρο είναι ικανή και αναγκαία συνθήκη για να υπάρχει ακμή ανάμεσα στους αντίστοιχους κόμβους στο άλλο δέντρο.

Θεωρείστε το ακόλουθο τμήμα ψευδοκώδικα για τον αλγόριθμο που υπολογίζει αν δύο δέντρα είναι ισομορφικά ή όχι.

```
1: IsoTree(x, y: Nodes): Boolean;  
2:   IF not (Size(x) = Size(y)) THEN return False; halt.  
3:   IF x = NIL THEN return True; halt.  
4:   IF IsoTree(Left(x), Left(y))  
5:     THEN  
6:       return IsoTree(Right(x), Right(y))  
7:     ELSE  
8:       return (IsoTree(Right(x), Left(y)) OR IsoTree(Left(x),  
Right(y)));
```

Θεωρείστε τα ακόλουθα δέντρα.



Λαμβάνοντας υπόψη την ακολουθία εκτέλεσης της κλήσης IsoTree(A, B), σε ποιές γραμμές (ή, ποιά γραμμή) του ψευδοκώδικα υπάρχουν λάθη, όσον αφορά στο αν γίνεται σωστά ο υπολογισμός του ισομορφισμού;

Τεκμηριώστε πολύ σύντομα την απάντησή σας.

| | | | |
|-------------------------------|----------------------------|----------------------------|----------------------------|
| <input type="checkbox"/> 6, 8 | <input type="checkbox"/> 6 | <input type="checkbox"/> 8 | <input type="checkbox"/> 3 |
|-------------------------------|----------------------------|----------------------------|----------------------------|

ΘΕΜΑ 17

Θεωρείστε μία παραλλαγή της υλοποίησης του Quicksort αλγόριθμου που διαλέγει πάντα το στοιχείο διαμέρισης x να είναι το $A[3]$, όπου $A[1...n]$ είναι ο πίνακας προς διαμέριση.

Τεκμηριώστε σύντομα αν αυτή η παραλλαγή τρέχει σε χρόνο $O(n \log n)$ στη χειρότερη περίπτωση χρησιμοποιώντας τον πίνακα εισόδου $\{n, n-1, 1, 2, 3, 4, \dots, n-2\}$ και υπολογίζοντας την αναδρομική έκφραση $T(n)$ για το χρόνο εκτέλεσης.

ΘΕΜΑ 18

Έχετε να τοποθετήσετε n βιβλία b_1, b_2, \dots, b_n , στα ράφια μιας βιβλιοθήκης με τη σειρά που σας δίνονται. Για κάθε $1 \leq i \leq n$ το βιβλίο b_i έχει πάχος t_i και ύψος h_i . Το μήκος κάθε ραφιού της βιβλιοθήκης είναι L . Υποθέστε πως τα ράφια έχουν προκαθορισμένο ύψος h και $h_i < h$ για κάθε i .



Ο παρακάτω άπληστος αλγόριθμος υπολογίζει την τοποθέτηση των βιβλίων που χρησιμοποιεί τον ελάχιστο δυνατό αριθμό ραφιών.

- (1) Ξεκινά με το πρώτο ράφι.
- (2) Εφόσον υπάρχει χώρος στο ράφι για το επόμενο στη σειρά βιβλίο, τότε τοποθέτησε το βιβλίο στο ράφι. Επανάλαβε το βήμα.
- (3) Αν δεν υπάρχει χώρος για το επόμενο βιβλίο, τότε δημιουργήσε ένα καινούργιο ράφι και επέστρεψε στο βήμα (2).

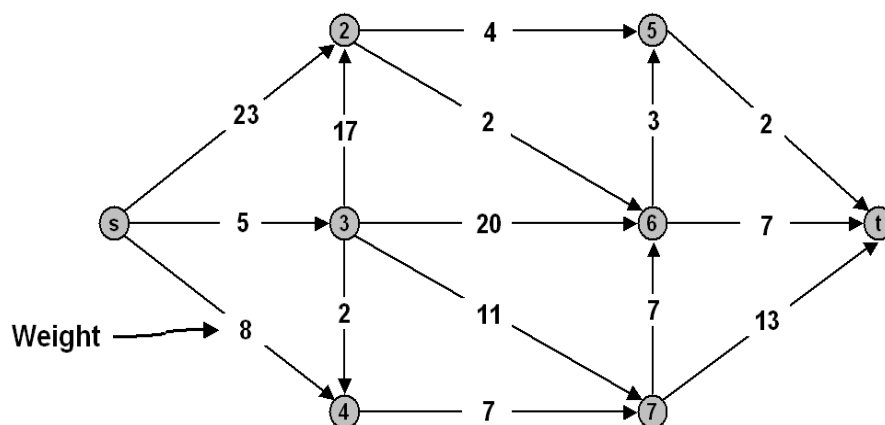
Δώστε την εκτίμησή σας για την πολυπλοκότητα του αλγορίθμου και τεκμηριώστε την.

Υποθέστε, τώρα, πως τα ράφια ΔΕΝ έχουν προκαθορισμένο ύψος, δηλαδή μπορείτε να αποφασίσετε εσείς το ύψος το καθενός από αυτά ανάλογα με το πιο ψηλό βιβλίο που περιέχει. Θέλετε να βρείτε τοποθέτηση βιβλίων που ελαχιστοποιεί το συνολικό ύψος των ραφιών που χρησιμοποιούνται.

Τεκμηριώστε αν ο προηγούμενος αλγόριθμος συνεχίζει να υπολογίζει την τοποθέτηση των βιβλίων που χρησιμοποιεί τον ελάχιστο δυνατό αριθμό ραφιών.

ΘΕΜΑ 19

Θεωρείστε τον παρακάτω κατευθυνόμενο γράφο:



Βρείτε το συντομότερο μονοπάτι (shortest path) από τον κόμβο s στον κόμβο t και υπολογίστε το μήκος του.

ΘΕΜΑ 20

Ως διάμετρος ενός γράφου ορίζεται η μεγαλύτερη απόσταση από όλες τις αποστάσεις συντομότερου μονοπατιού μεταξύ δύο οποιονδήποτε κόμβων του γράφου αυτού. Σε ένα γράφο χωρίς βάρη στις ακμές του, η απόσταση μεταξύ δύο κόμβων καθορίζεται από το πλήθος των ακμών μεταξύ των κόμβων αυτών. Δώστε την περιγραφή ενός αλγόριθμου ο οποίος θα υπολογίζει τη διάμετρο ενός συνεκτικού και χωρίς βάρη στις ακμές του γράφου σε χρόνο $O(V \cdot E)$, όπου V είναι ο αριθμός των κόμβων και E είναι ο αριθμός των ακμών του γράφου. Αποδείξτε την πολυπλοκότητα χρόνου του αλγόριθμου που προτείνεται.

ΘΕΜΑ 21

Έστω οι παρακάτω συναρτήσεις:

```
f1(n)
1. while n > 1
```



```
2. do n ← n - 1
```

```
f2(n)
```

```
1. while n > 1
```

```
2. do n ← ⌊n/2⌋
```

```
f3(n)
```

```
1. while n > 1
```

```
2. do for i ← 1 to n
```

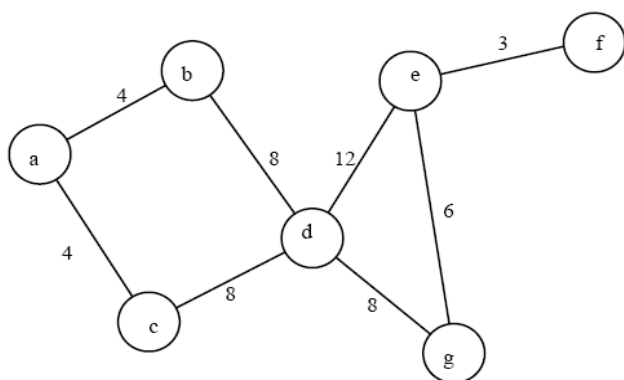
```
3. do j ← 1
```

```
4. n ← n - 1
```

Θεωρείστε ότι οι παραπάνω συναρτήσεις καλούνται με όρισμα μόνο κάποιο θετικό ακέραιο $n > 0$. Ποια είναι η πολυπλοκότητα ως προς το n για κάθε μια από τις παραπάνω συναρτήσεις;

ΘΕΜΑ 22

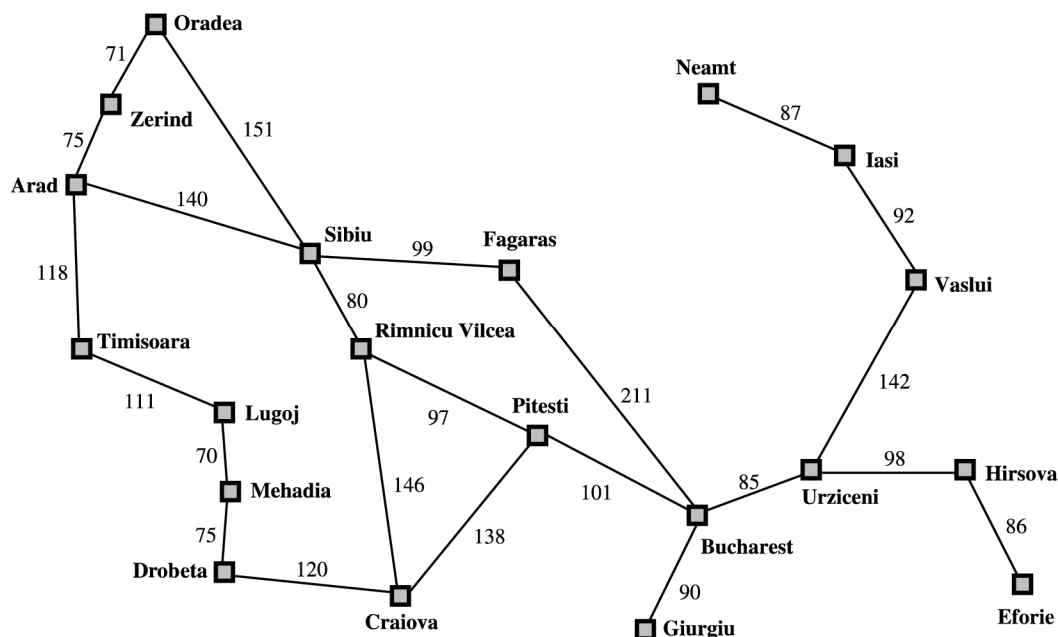
Έστω ο παρακάτω μη κατευθυνόμενος γράφος:



Κατασκευάστε ένα minimum spanning tree για το γράφο αυτό.

ΘΕΜΑ 23

Δίνεται ο παρακάτω γράφος που αφορά αποστάσεις μεταξύ πόλεων της Ρουμανίας.



Βρείτε τη συντομότερη διαδρομή από το Bucharest προς την Timisoara χρησιμοποιώντας κάποιο συγκεκριμένο αλγόριθμο. [Σημείωση: Ο χάρτης είναι από το βιβλίο Artificial Intelligence, A modern approach]

ΘΕΜΑ 24

Έστω το παρακάτω σύνολο διαλέξεων που πρέπει να πραγματοποιηθούν κατά τη διάρκεια της ίδιας ημέρας εντός ενός ακαδημαϊκού ιδρύματος. Για κάθε διάλεξη δίνεται ο χρόνος έναρξης και λήξης αυτής:

| Διάλεξη | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---------|----|----|----|----|----|----|----|----|----|----|----|
| Έναρξη | 9 | 11 | 8 | 13 | 11 | 13 | 14 | 16 | 16 | 10 | 20 |
| Λήξη | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |

Κάθε διάλεξη απαιτεί τη δέσμευση μιας αίθουσας του ιδρύματος για να πραγματοποιηθεί. Δεν επιτρέπεται να πραγματοποιούνται ταυτόχρονα δύο διαφορετικές διαλέξεις στην ίδια αίθουσα, όπως επίσης δεν επιτρέπεται ένα τμήμα μιας διάλεξης να πραγματοποιηθεί σε μία αίθουσα και ένα άλλο τμήμα της ίδιας διάλεξης να πραγματοποιηθεί σε άλλη αίθουσα. Βρείτε τον ελάχιστο αριθμό αιθουσών που πρέπει να διαθέτει το ίδρυμα ώστε να είναι δυνατή η πραγματοποίηση όλων των διαλέξεων.

| | | | |
|----------------------------|----------------------------|----------------------------|----------------------------|
| <input type="checkbox"/> 3 | <input type="checkbox"/> 4 | <input type="checkbox"/> 5 | <input type="checkbox"/> 6 |
|----------------------------|----------------------------|----------------------------|----------------------------|

ΘΕΜΑ 25

Έστω δύο διασυνδεδεμένες λίστες L1 και L2 οι οποίες έχουν ένα κοινό μέρος μεγέθους k.

Το μήκος των L1 και L2 μέχρι τον 1^ο κοινό κόμβο είναι n και m αντίστοιχα.

Προτείνεται έναν αλγόριθμο που θα βρίσκει τον 1^ο κοινό κόμβο σε χρόνο $O(n+m+k)$.



Κριτήρια αξιολόγησης:

Το κάθε ένα από τα παραπάνω 25 θέματα λαμβάνει 4 μονάδες (σύνολο 100μ). Σημειώστε ότι οι επιμέρους μονάδες δεν αντανakλούν τη δυσκολία των θεμάτων.

Ο συνολικός βαθμός θα διαιρεθεί δια 10, ώστε να προκύψει ο τελικός βαθμός της εργασίας (με μέγιστη τιμή το 10).

Τρόπος – Ημερομηνία Παράδοσης

Η εργασία σας θα πρέπει να έχει φτάσει στον Καθηγητή-Σύμβουλό σας μέχρι την Κυριακή 18/5/2008 ώρα 23:59.

Περιμένουμε όλες οι εργασίες να αποσταλούν μέσω Email και να είναι γραμμένες σε επεξεργαστή κειμένου MSWord (διευκρινήσεις, σχόλια και παραδοχές που θεωρείτε απαραίτητα να συμπεριλάβετε στην εργασία σας). Στον Καθηγητή-Σύμβουλό σας, σε κάθε περίπτωση, στέλνετε ΕΝΑ μόνο αρχείο (συμπιεσμένο).

Δεν θα δοθεί παράταση στην παράδοση της εργασίας πέραν της ως άνω αναφερόμενης ημέρας και ώρας, για κανένα λόγο. Την Τρίτη 20/5/2008 και ώρα 13:00, θα δημοσιευθεί ενδεικτική επίλυση της εργασίας στο διαδίκτυο (portal).

Καλή Επιτυχία!!!