

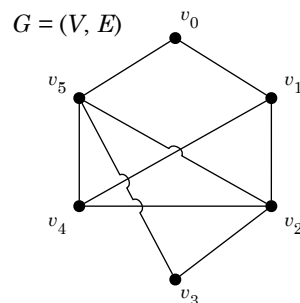
Seconda prova di esonero – 11 gennaio 2022

Risolvere i seguenti problemi proponendo, per ciascun esercizio, la codifica in linguaggio C di un programma completo. La prova dura tre ore, durante le quali non è possibile allontanarsi dall'aula, se non dopo aver consegnato l'elaborato scritto. Per superare la prova di esonero è necessario ottenere almeno 15 punti; tuttavia affinché le prove di esonero siano valide è necessario che la media dei voti del primo e del secondo esonero sia maggiore o uguale a 18/30. È possibile utilizzare libri e appunti personali, senza scambiarli con altri studenti. I compiti che presenteranno evidenti ed anomale "similitudini" saranno annullati.

Esercizio n. 1

Lette in input le liste di adiacenza di un grafo non orientato $G = (V, E)$, visualizzare in output, se esistono, tutte le coppie di vertici $u, v \in V(G)$ tali che u domina v (il vertice u domina il vertice v se $N[v] \setminus \{u\} \subseteq N[u]$, ossia se u è adiacente a tutti i vertici a cui è adiacente anche v).

Esempio Si consideri il grafo G rappresentato in figura. Il vertice v_5 domina il vertice v_1 , infatti $N[v_1] = \{v_0, v_2, v_4\} \subset N[v_5] = \{v_0, v_2, v_3, v_4\}$, v_5 domina v_3 , v_4 domina v_0 e v_3 .



Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 struct nodo {
6     int info;
7     struct nodo *next;
8 };
9

```

```

10 struct nodo *leggiLista(void) {
11     struct nodo *p, *primo = NULL;
12     int i, n;
13     printf("Numero di elementi: ");
14     scanf("%d", &n);
15     printf("Inserisci %d vertici: ", n);
16     for (i=0; i<n; i++) {
17         p = malloc(sizeof(struct nodo));
18         p->next = primo;
19         scanf("%d", &p->info);
20         primo = p;
21     }
22     return(primo);
23 }
24
25 int leggiGrafo(struct nodo *V[]) {
26     int i, n;
27     printf("Numero di vertici del grafo: ");
28     scanf("%d", &n);
29     for (i=0; i<n; i++) {
30         printf("Lista di adiacenza di %d\n", i);
31         V[i] = leggiLista();
32     }
33     return(n);
34 }
35
36 int domina(struct nodo *V[], int u, int v) {
37     int r = 1;
38     struct nodo *p, *q;
39     p = V[v];
40     while (p != NULL && r == 1) {
41         q = V[u];
42         r = 0;
43         while (q != NULL && r == 0) {
44             if (q->info == p->info)
45                 r = 1;
46             q = q->next;
47         }
48         p = p->next;
49     }
50     return(r);
51 }
52
53 int main(void) {
54     struct nodo *G[MAX];
55     int n, i, j;
56     n = leggiGrafo(G);

```

```
57  for (i = 0; i < n; i++)
58      for (j = 0; j < n; j++)
59          if (i != j && domina(G, i, j))
60              printf("%d domina %d\n", i, j);
61  return(0);
62 }
```

Esercizio n. 2

Generare una lista di n numeri casuali interi maggiori di zero e minori di 10, $L : v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_{n-1}$. Visualizzare in output la lista. Partendo dal primo elemento della lista, v_0 , visualizzare iterativamente l'elemento v_i e poi l'elemento che si trova v_i posizioni dopo nella lista, fino al termine della lista.

Esempio Sia $n = 11$ e supponiamo che la lista L sia la seguente:

$3 \rightarrow 2 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow \text{NULL}$

Allora viene stampata in output la sequenza 3, 2, 2, 8.

Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4
5 struct nodo {
6     int info;
7     struct nodo *next;
8 };
9
10 void stampaLista(struct nodo *p) {
11     while (p != NULL) {
12         printf("%d --> ", p->info);
13         p = p->next;
14     }
15     printf("NULL\n");
16     return;
17 }
18
19 struct nodo *listaCasuale(void) {
20     struct nodo *primo = NULL, *p;
21     int i, n;
22     printf("Numero di elementi: ");
23     scanf("%d", &n);
24     srand((unsigned)time(NULL));
25     for (i=0; i<n; i++) {
26         p = malloc(sizeof(struct nodo));
27         p->info = rand() % 9 + 1;
28         p->next = primo;
29         primo = p;
30     }
31     return(primo);
32 }
```

```

33
34 void stampaElementi(struct nodo *p) {
35     struct nodo *q;
36     int i;
37     do {
38         printf("%d ", p->info);
39         q = p->next;
40         i = 1;
41         while (q != NULL && i < p->info) {
42             q = q->next;
43             i++;
44         }
45         p = q;
46     } while (p != NULL);
47     printf("\n");
48     return;
49 }
50
51 int main(void) {
52     struct nodo *p;
53     p = listaCasuale();
54     stampaLista(p);
55     stampaElementi(p);
56     return(0);
57 }

```