

Corso di Algoritmi e Strutture Dati (IN110)

Tutorato n. 9

Marco Liverani*

Esercizio n. 1

Leggere in input un grafo orientato $G = (V, E)$ e rappresentarlo mediante liste di adiacenza. Stampare le liste di adiacenza del grafo. Letto in input un vertice $v \in V(G)$ ($0 \leq v \leq n$) stampare tutti i vertici $u \in V(G)$ a cui v è adiacente (tutti i vertici u tali che $(u, v) \in E(G)$).

Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 struct nodo {
5     int info;
6     struct nodo *next;
7 };
8
9 struct nodo *leggi_lista(void) {
10     struct nodo *p, *primo;
11     int i, n;
12     printf(" inserisci il numero di elementi: ");
13     scanf("%d", &n);
14     printf(" inserisci %d elementi: ", n);
15     primo = NULL;
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 void stampa_lista(struct nodo *p) {
26     while (p != NULL) {
27         printf("%d --> ", p->info);
28         p = p->next;
29     }
```

*Università degli Studi Roma Tre, Corso di Laurea in Matematica, Corso di Algoritmi e Strutture Dati (IN110); e-mail liverani@mat.uniroma3.it – sito web del corso <http://www.mat.uniroma3.it/users/liverani/IN110/>

```

30     printf("Null\n");
31     return;
32 }
33
34 int leggi_grafo(struct nodo *G[]) {
35     int i, n;
36     printf("Inserisci il numero di vertici del grafo: ");
37     scanf("%d", &n);
38     for (i=0; i<n; i++) {
39         printf("Lista di adiacenza del vertice %d:\n", i);
40         G[i] = leggi_lista();
41     }
42     return(n);
43 }
44
45 void stampa_grafo(struct nodo *G[], int n) {
46     int i;
47     printf("Liste di adiacenza dei vertici del grafo:\n");
48     for (i=0; i<n; i++) {
49         printf(" vertici adiacenti a %d: ", i);
50         stampa_lista(G[i]);
51     }
52     return;
53 }
54
55 int adiacente(int a, int b, struct nodo *G[]) {
56     struct nodo *p;
57     int r;
58     p = G[a];
59     while (p != NULL && p->info != b)
60         p = p->next;
61     if (p == NULL)
62         r = 0;
63     else
64         r = 1;
65     return(r);
66 }
67
68 int main(void) {
69     struct nodo *L[20];
70     int n, u, v;
71     n = leggi_grafo(L);
72     stampa_grafo(L, n);
73     printf("Inserisci un vertice del grafo: ");
74     scanf("%d", &v);
75     printf("Vertici a cui %d e' adiacente: ", v);
76     for (u=0; u<n; u++)
77         if (adiacente(u, v, L))
78             printf("%d ", u);
79     printf("\n");
80     return(0);
81 }

```

Esercizio n. 2

Leggere in input un grafo orientato $G = (V, E)$ e rappresentarlo mediante liste di adiacenza. Stampare le liste di adiacenza del grafo. Letto in input un vertice v ($0 \leq v \leq n$) eliminare tutti gli spigoli entranti in v . Stampare le liste di adiacenza del grafo.

Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 struct nodo {
5     int info;
6     struct nodo *next;
7 };
8
9 struct nodo *leggi_lista(void) {
10     struct nodo *p, *primo;
11     int i, n;
12     printf(" inserisci il numero di elementi: ");
13     scanf("%d", &n);
14     printf(" inserisci %d elementi: ", n);
15     primo = NULL;
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 void stampa_lista(struct nodo *p) {
26     while (p != NULL) {
27         printf("%d --> ", p->info);
28         p = p->next;
29     }
30     printf("Null\n");
31     return;
32 }
33
34 int leggi_grafo(struct nodo *G[]) {
35     int i, n;
36     printf("Inserisci il numero di vertici del grafo: ");
37     scanf("%d", &n);
38     for (i=0; i<n; i++) {
39         printf("Lista di adiacenza del vertice %d:\n", i);
40         G[i] = leggi_lista();
41     }
42     return(n);
43 }
44
45
```

```

46 void stampa_grafo(struct nodo *G[], int n) {
47     int i;
48     printf("Liste di adiacenza dei vertici del grafo:\n");
49     for (i=0; i<n; i++) {
50         printf(" vertici adiacenti a %d: ", i);
51         stampa_lista(G[i]);
52     }
53     return;
54 }
55
56 void elimina_spigoli(struct nodo *G[], int n, int v) {
57     int i;
58     struct nodo *p, *prec;
59     for (i=0; i<n; i++) {
60         p = G[i];
61         prec = NULL;
62         while (p != NULL && p->info != v) {
63             prec = p;
64             p = p->next;
65         }
66         if (p != NULL) {
67             if (prec == NULL) {
68                 G[i] = p->next;
69                 free(p);
70             } else {
71                 prec->next = p->next;
72                 free(p);
73             }
74         }
75     }
76     return;
77 }
78
79 int main(void) {
80     struct nodo *L[20];
81     int n, v;
82     n = leggi_grafo(L);
83     stampa_grafo(L, n);
84     printf("Inserisci un vertice del grafo: ");
85     scanf("%d", &v);
86     elimina_spigoli(L, n, v);
87     stampa_grafo(L, n);
88     return(0);
89 }

```

Esercizio n. 3

Leggere in input una sequenza di n numeri *floating point* e rappresentarla con una lista L ; stampare la lista. Costruire una seconda lista L' composta dai soli elementi di L maggiori della media; stampare L' .

Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 struct nodo {
5     float info;
6     struct nodo *next;
7 };
8
9 struct nodo *leggi_lista(void) {
10     struct nodo *p, *primo;
11     int i, n;
12     printf(" inserisci il numero di elementi: ");
13     scanf("%d", &n);
14     printf(" inserisci %d elementi: ", n);
15     primo = NULL;
16     for (i=0; i<n; i++) {
17         p = malloc(sizeof(struct nodo));
18         p->next = primo;
19         scanf("%f", &p->info);
20         primo = p;
21     }
22     return(primo);
23 }
24
25 void stampa_lista(struct nodo *p) {
26     while (p != NULL) {
27         printf("%3.2f --> ", p->info);
28         p = p->next;
29     }
30     printf("Null\n");
31     return;
32 }
33
34 float media(struct nodo *p) {
35     float s=0.0, n=0.0;
36     while (p != NULL) {
37         s = s + p->info;
38         n = n + 1;
39         p = p->next;
40     }
41     return(s/n);
42 }
43
44
45
46
```

```
47 int main(void) {
48     struct nodo *p, *q, *r, *s;
49     float m;
50     p = leggi_lista();
51     stampa_lista(p);
52     m = media(p);
53     r = NULL;
54     q = p;
55     while (q != NULL) {
56         if (q->info > m) {
57             s = malloc(sizeof(struct nodo));
58             s->info = q->info;
59             s->next = r;
60             r = s;
61         }
62         q = q->next;
63     }
64     stampa_lista(r);
65     return(0);
66 }
```