

Corso di Algoritmi e Strutture Dati (IN110) – Prof. Marco Liverani – a.a. 2023/2024

Esame scritto del 13 Settembre 2024 (Appello X)

Si richiede di risolvere entrambi gli esercizi riportando una codifica in linguaggio C completa dei due programmi. Nel caso in cui non si riesca a completare entrambi gli esercizi si suggerisce di riportare almeno la codifica in C delle funzioni principali o una loro pseudo-codifica. È possibile consultare libri e appunti personali, ma non scambiare libri o appunti con altri studenti. I compiti che presenteranno evidenti ed anomale «similitudini» saranno annullati. La prova scritta ha una durata di tre ore, durante le quali non è consentito allontanarsi dall'aula, se non dopo aver consegnato il compito.

Deve essere consegnata solo la «bella copia» del compito scritto; su ciascun foglio deve essere riportato il **nome**, il **cognome** e il **numero di matricola** dello studente.

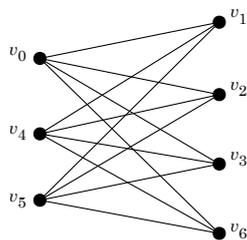
Esercizio n. 1

Letto in input un intero $n > 0$, costruire un array $A = (a_1, \dots, a_{n-1})$ di n numeri casuali in $\{0, 1\}$. Costruire le liste di adiacenza del grafo bipartito completo $G = (V, E)$ con $V = \{0, \dots, n-1\}$ e $(v_i, v_j) \in E$ se e solo se $a_i \neq a_j$. Stampare il vettore A e le liste di adiacenza del grafo G .

Esempio Sia $n = 7$ e si consideri il seguente array di numeri casuali:

$$A = (a_0 = 0, a_1 = 1, a_2 = 1, a_3 = 1, a_4 = 0, a_5 = 0, a_6 = 1)$$

Il grafo bipartito completo G costruito secondo le indicazioni del testo dell'esercizio è il seguente



Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 50
5
6 struct nodo {
7     int info;
8     struct nodo *next;
9 };
10
11 void arrayCasuale(int A[], int n) {
12     srand((unsigned)time(NULL));
13     for (int i=0; i<n; i++)
14         A[i] = rand() % 2;
15     return;
16 }

```

```

17
18 void grafoBipartito(struct nodo *V[], int A[], int n) {
19     int i, j;
20     struct nodo *p;
21     for (i=0; i<n; i++)
22         V[i] = NULL;
23     for (i=0; i<n-1; i++) {
24         for (j=i; j<n; j++) {
25             if (A[i] != A[j]) {
26                 p = malloc(sizeof(struct nodo));
27                 p->info = j;
28                 p->next = V[i];
29                 V[i] = p;
30                 p = malloc(sizeof(struct nodo));
31                 p->info = i;
32                 p->next = V[j];
33                 V[j] = p;
34             }
35         }
36     }
37     return;
38 }
39
40 void stampaLista(struct nodo *p) {
41     while (p != NULL) {
42         printf("%d --> ", p->info);
43         p = p->next;
44     }
45     printf("Null\n");
46     return;
47 }
48
49 void stampaGrafo(struct nodo *V[], int n) {
50     for (int i=0; i<n; i++) {
51         printf("%2d: ", i);
52         stampaLista(V[i]);
53     }
54     return;
55 }
56
57 void stampaArray(int A[], int n) {
58     for (int i=0; i<n; i++)
59         printf("%d ", A[i]);
60     printf("\n");
61     return;
62 }
63
64 int main(void) {
65     int n, A[MAX];
66     struct nodo *V[MAX];
67     printf("Inserisci n: ");
68     scanf("%d", &n);
69     arrayCasuale(A, n);
70     grafoBipartito(V, A, n);
71     stampaArray(A, n);
72     stampaGrafo(V, n);

```

```
73 | return 0;
74 | }
```

Esercizio n. 2

Letti in input due interi $n, m > 0$, costruire una matrice A di $n \times m$ numeri casuali compresi in $\{0, 1\}$. Stampare la matrice. Letto in input un intero k , $0 \leq k \leq m$, stampare l'indice h di una colonna di A con il massimo numero di elementi in comune con la colonna di indice k ($h \neq k$).

Esempio Siano $n = 6$, $m = 8$ e $k = 2$. Si consideri la seguente matrice A :

$$A = \begin{pmatrix} 1 & 0 & \mathbf{1} & 1 & 1 & 1 & 0 & \mathbf{1} \\ 1 & 0 & \mathbf{0} & 0 & 0 & 1 & 1 & \mathbf{0} \\ 1 & 1 & \mathbf{0} & 1 & 0 & 1 & 1 & \mathbf{0} \\ 1 & 0 & \mathbf{1} & 0 & 0 & 0 & 1 & \mathbf{1} \\ 1 & 0 & \mathbf{1} & 1 & 0 & 0 & 1 & \mathbf{0} \\ 0 & 0 & \mathbf{0} & 0 & 1 & 1 & 0 & \mathbf{0} \end{pmatrix}$$

In questo esempio, l'ultima colonna (quella di indice $h = 7$) è la colonna con più elementi uguali alla terza colonna (quella con indice $k = 2$).

Soluzione

```
1 | #include <stdlib.h>
2 | #include <stdio.h>
3 | #include <time.h>
4 | #define MAX 50
5 |
6 | void matriceCasuale(int A[MAX][MAX], int n, int m) {
7 |     srand((unsigned)time(NULL));
8 |     for (int i=0; i<n; i++)
9 |         for (int j=0; j<m; j++)
10 |             A[i][j] = rand() % 2;
11 |     return;
12 | }
13 |
14 | int colonneSimili(int A[MAX][MAX], int n, int m, int k) {
15 |     int h, max=0, cont, i, j;
16 |     for (j=0; j<m; j++)
17 |         if (j != k) {
18 |             cont = 0;
19 |             for (i=0; i<n; i++)
20 |                 if (A[i][k] == A[i][j])
21 |                     cont++;
22 |             if (cont > max) {
23 |                 h = j;
24 |                 max = cont;
25 |             }
26 |         }
27 |     return(h);
28 | }
29 |
30 | void stampaMatrice(int A[MAX][MAX], int n, int m) {
31 |     for (int i=0; i<n; i++) {
32 |         for (int j=0; j<m; j++)
```

```
33     printf("%d ", A[i][j]);
34     printf("\n");
35 }
36 return;
37 }
38
39 int main(void) {
40     int m, n, h, k, A[MAX][MAX];
41     printf("Inserisci tre interi positivi n, m e k: ");
42     scanf("%d %d %d", &n, &m, &k);
43     matriceCasuale(A, n, m);
44     stampaMatrice(A, n, m);
45     h = colonneSimili(A, n, m, k);
46     printf("Colonna piu' simile alla %d: %d\n", k, h);
47     return 0;
48 }
```