

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

Esame scritto del 14 Giugno 2024 (Appello C)

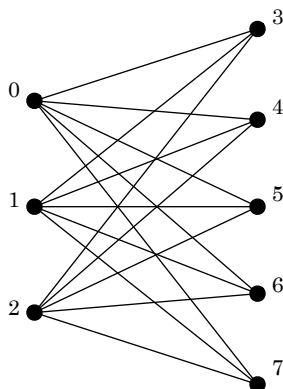
Si richiede di risolvere entrambi gli esercizi riportando una codifica in linguaggio C completa dei due programmi. Nel caso in cui non si riesca a completare entrambi gli esercizi si suggerisce di riportare almeno la codifica in C delle funzioni principali o una loro pseudo-codifica. È possibile consultare 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 (o un altro codice identificativo di fantasia) dello studente.

Esercizio n. 1

Letti in input due numeri interi positivi $n, h > 0$ con $n > h$, costruire le liste di adiacenza del grafo bipartito completo $K_{h,n-h} = (V_1 \cup V_2, E)$, tale che $V_1 = \{0, 1, \dots, h-1\}$, $V_2 = \{h, h+1, \dots, n-1\}$ ed $E = \{(u, v) \text{ t.c. } u \in V_1, v \in V_2\}$. Stampare le liste di adiacenza di $K_{h,n-h}$.

Esempio Siano $n = 8, h = 3$. Il grafo bipartito completo $K_{3,5}$ costruito secondo le indicazioni del testo dell’esercizio è il seguente



Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 50
4
5 struct nodo {
6     int info;
7     struct nodo *next;
8 };
9
10 void grafoBipartitoCompleto(struct nodo *V[], int n, int h) {
11     int i, j;
12     struct nodo *p;
13     for (i=0; i<n; i++)
    
```

```

14     V[i] = NULL;
15     for (i=0; i<h; i++) {
16         for (j=h; j<n; j++) {
17             p = malloc(sizeof(struct nodo));
18             p->info = j;
19             p->next = V[i];
20             V[i] = p;
21             p = malloc(sizeof(struct nodo));
22             p->info = i;
23             p->next = V[j];
24             V[j] = p;
25         }
26     }
27     return;
28 }
29
30 void stampaLista(struct nodo *p) {
31     while (p != NULL) {
32         printf("%d --> ", p->info);
33         p = p->next;
34     }
35     printf("Null\n");
36     return;
37 }
38
39 void stampaGrafo(struct nodo *V[], int n) {
40     for (int i=0; i<n; i++) {
41         printf("%2d: ", i);
42         stampaLista(V[i]);
43     }
44     return;
45 }
46
47 int main(void) {
48     int n, h;
49     struct nodo *V[MAX];
50     printf("Inserisci n e h (n>h) positivi: ");
51     scanf("%d %d", &n, &h);
52     grafoBipartitoCompleto(V, n, h);
53     stampaGrafo(V, n);
54     return 0;
55 }

```

Esercizio n. 2

Letti in input due interi $n, k > 0$, costruire un vettore A di numeri casuali compresi in $\{1, 2, \dots, k\}$. Stampare il vettore A . Costruire una matrice quadrata M di ordine n (con n righe ed n colonne) tale che la riga di indice i di M sia composta dagli elementi $a_i, a_{i+1}, \dots, a_{n-1}, a_0, a_1, \dots, a_{i-1}$ di A . Stampare in output la matrice M .

Esempio Siano $n = 6, k = 10$. Si consideri il seguente vettore $A = (7, 1, 3, 7, 5, 10)$ di numeri casuali; in questo caso la matrice M è la seguente:

$$\begin{pmatrix} 7 & 1 & 3 & 7 & 5 & 10 \\ 1 & 3 & 7 & 5 & 10 & 7 \\ 3 & 7 & 5 & 10 & 7 & 1 \\ 7 & 5 & 10 & 7 & 1 & 3 \\ 5 & 10 & 7 & 1 & 3 & 7 \\ 10 & 7 & 1 & 3 & 7 & 5 \end{pmatrix}$$

Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 50
5
6 void arrayCasuale(int A[], int n, int k) {
7     srand((unsigned)time(NULL));
8     for (int i=0; i<n; i++)
9         A[i] = rand() % k + 1;
10    return;
11 }
12
13 void costruisciMatrice(int M[MAX][MAX], int A[], int n) {
14     for (int i=0; i<n; i++)
15         for (int j=0; j<n; j++)
16             M[i][j] = A[(j+i) % n];
17    return;
18 }
19
20 void stampaArray(int A[], int n) {
21     for (int i=0; i<n; i++)
22         printf("%2d ", A[i]);
23     printf("\n");
24    return;
25 }
26
27 void stampaMatrice(int M[MAX][MAX], int n) {
28     for (int i=0; i<n; i++) {
29         for (int j=0; j<n; j++)
30             printf("%2d ", M[i][j]);
31         printf("\n");
32     }
33    return;
34 }
35
36 int main(void) {
37     int k, n, A[MAX], M[MAX][MAX];
```

```
38 | printf("Inserisci due interi positivi n e k: ");
39 | scanf("%d %d", &n, &k);
40 | arrayCasuale(A, n, k);
41 | stampaArray(A, n);
42 | costruisciMatrice(M, A, n);
43 | stampaMatrice(M, n);
44 | return 0;
45 | }
```