

# Corso di Algoritmi e Strutture Dati (IN110)

## Tutorato n. 6

Marco Liverani\*

### Esercizio n. 1

Letti in input due interi positivi  $n$  e  $m$ , genera in modo casuale una matrice di ordine  $n \times m$  di numeri interi minori di 10. Stampa le colonne che contengono tre elementi consecutivi che abbiano valori successivi.

#### Pseudo-codifica dell'algoritmo

- 1: leggi  $n$  e  $m$
- 2: per ogni  $i = 0, 1, \dots, n - 1$  e per ogni  $j = 0, 1, \dots, m - 1$  poni  $A_{i,j}$  = intero casuale compreso tra 0 e 10
- 3: stampa la matrice  $A$
- 4: **per**  $j = 0, 1, \dots, m - 1$  **ripeti**
- 5: verifica se nella colonna  $j$  della matrice  $A$  sono presenti tre elementi  $A_{i,j}, A_{i+1,j}, A_{i+2,j}$  tali che  
 $A_{i,j} = A_{i+1,j} - 1 = A_{i+2,j} - 2$
- 6: se la condizione è verificata stampa la colonna  $j$  della matrice  $A$
- 7: **fine-ciclo**
- 8: stop

#### Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 100
5
6 void stampa_colonna(int x[MAX][MAX], int c, int n) {
7     int i;
8     for (i=0; i<n; i++)
9         printf("%d ", x[i][c]);
10    printf("\n");
11    return;
12 }
13
14 void stampa_matrice(int x[MAX][MAX], int n, int m) {
15     int i, j;
16     for (i=0; i<n; i++) {
17         for (j=0; j<m; j++)
18             printf("%4d ", x[i][j]);
```

\*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/>

```

19     printf("\n");
20 }
21 return;
22 }
23
24 void genera_matrice(int x[MAX][MAX], int n, int m) {
25     int i, j;
26     srand((unsigned)time(NULL));
27     for (i=0; i<n; i++)
28         for (j=0; j<m; j++)
29             x[i][j] = rand()%10;
30     return;
31 }
32
33 int verifica(int x[MAX][MAX], int c, int n) {
34     int i, r=0;
35     for (i=0; i<n-2 && r==0; i++) {
36         if (x[i][c] == x[i+1][c]-1 && x[i+1][c] == x[i+2][c]-1)
37             r = 1;
38     }
39     return(r);
40 }
41
42 int main(void) {
43     int i, n, m, mat[MAX][MAX];
44     printf("Numero di righe e di colonne: ");
45     scanf("%d %d", &n, &m);
46     genera_matrice(mat, n, m);
47     stampa_matrice(mat, n, m);
48     for (i=0; i<m; i++) {
49         if (verifica(mat, i, n))
50             stampa_colonna(mat, i, n);
51     }
52     return(1);
53 }

```

## Esercizio n. 2

Letti in input due numeri interi positivi  $n$  e  $m$ , costruire un vettore  $A$  di numeri casuali tali che  $0 < a_0 < 10$  e  $a_{i-1} \leq a_i < 2a_{i-1}$  per ogni  $i = 1, 2, \dots, n-1$ . Stampare il vettore  $A$ . Generare in modo casuale un secondo array  $B$  di  $m$  numeri interi positivi minori di  $a_{n-1}$  e stampare  $B$ . Inserire gli elementi di  $B$  nella posizione corretta nell'array  $A$  in modo che  $A$  continui ad essere ordinato, eventualmente spostando in avanti gli elementi già presenti per fare posto ai nuovi elementi da inserire. Stampare in output l'array  $A$ .

### Pseudo-codifica dell'algoritmo

- 1: genera l'array  $A$  di  $n$  elementi tali che  $a_0 = (\text{rand mod } 9) + 1$  e  $a_i = (\text{rand mod } a_{i-1}) + a_{i-1}$  per  $i = 1, 2, \dots, n-1$
- 2: genera l'array  $B$  di  $m$  elementi casuali tali che  $B_i = \text{rand mod } a_{n-1}$  per ogni  $i = 0, 1, \dots, m-1$
- 3: stampa  $A$  e  $B$
- 4: **per**  $j = 0, 1, \dots, m-1$  **ripeti**
- 5:      $i = n-1 + j$
- 6:     **fintanto che**  $i \geq 0$  e  $a_i > b_j$  **ripeti**
- 7:          $a_{i+1} = a_i$
- 8:          $i = i - 1$
- 9:     **fine-ciclo**
- 10:      $a_{i+1} = b_j$
- 11: **fine-ciclo**
- 12: stampa  $A$
- 13: fermati

### Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 100
5
6 int genera_array_crescente(int V[]) {
7     int i, n;
8     printf("Numero di elementi: ");
9     scanf("%d", &n);
10    V[0] = rand() % 9 + 1;
11    for (i=1; i<n; i++)
12        V[i] = rand() % V[i-1] + V[i-1];
13    return(n);
14 }
15
16 int genera_array(int V[], int x) {
17     int i, n;
18     printf("Numero di elementi: ");
19     scanf("%d", &n);
20     for (i=0; i<n; i++)
21         V[i] = rand() % x;
22     return(n);
23 }
24
25 void stampa_array(int V[], int n) {
26     int i;
```

```

27     for (i=0; i<n; i++)
28         printf("%d ", V[i]);
29     printf("\n");
30     return;
31 }
32
33 void inserisci(int A[], int n, int B[], int m) {
34     int i, j;
35     for (j=0; j<m; j++) {
36         i = n+j-1;
37         while (i>=0 && A[i]>B[j]) {
38             A[i+1] = A[i];
39             i--;
40         }
41         A[i+1] = B[j];
42     }
43     return;
44 }
45
46 int main(void) {
47     int n, m, A[MAX], B[MAX];
48     srand((unsigned)time(NULL));
49     n = genera_array_crescente(A);
50     m = genera_array(B, A[n-1]);
51     printf("A = ");
52     stampa_array(A, n);
53     printf("B = ");
54     stampa_array(B, m);
55     inserisci(A, n, B, m);
56     printf("Nuovo A = ");
57     stampa_array(A, n+m);
58     return(1);
59 }

```