

# Corso di Algoritmi e Strutture Dati (IN110)

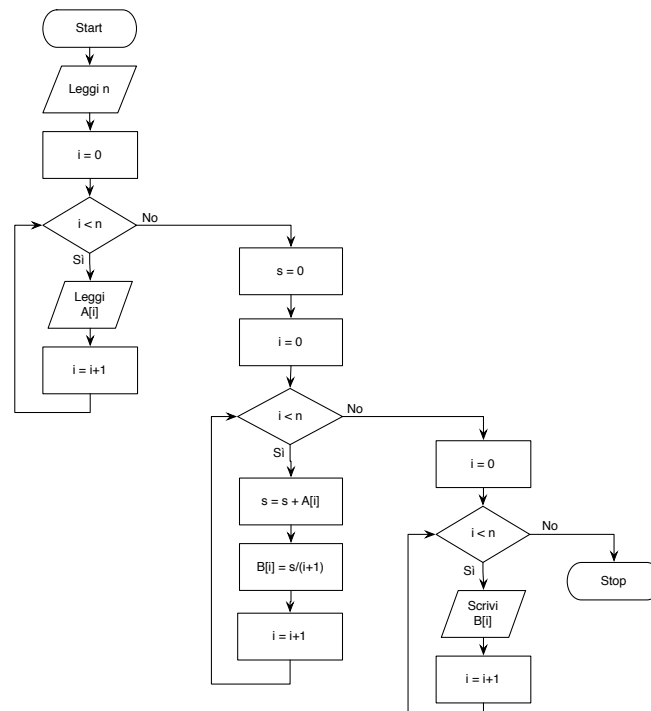
## Tutorato n. 2

Marco Liverani\*

### Esercizio n. 1

Letto in input un array  $A$  di  $n$  interi, costruire un array  $B$  di  $n$  numeri tale che  $B_i$  sia dato dalla media aritmetica dei primi  $i$  elementi di  $A$ . Stampare il vettore  $B$ .

### Diagramma di flusso



\*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/>

## Pseudo-codifica dell'algoritmo

```
1: leggi  $n$ 
2: leggi  $n$  numeri interi e memorizzali nell'array  $A$ 
3:  $s = 0$ 
4: per  $i = 0, 1, \dots, n-1$  ripeti
5:    $s = s + A_i$ 
6:    $B_i = \frac{s}{i+1}$ 
7: fine-ciclo
8: stampa l'array  $B$ 
9: stop
```

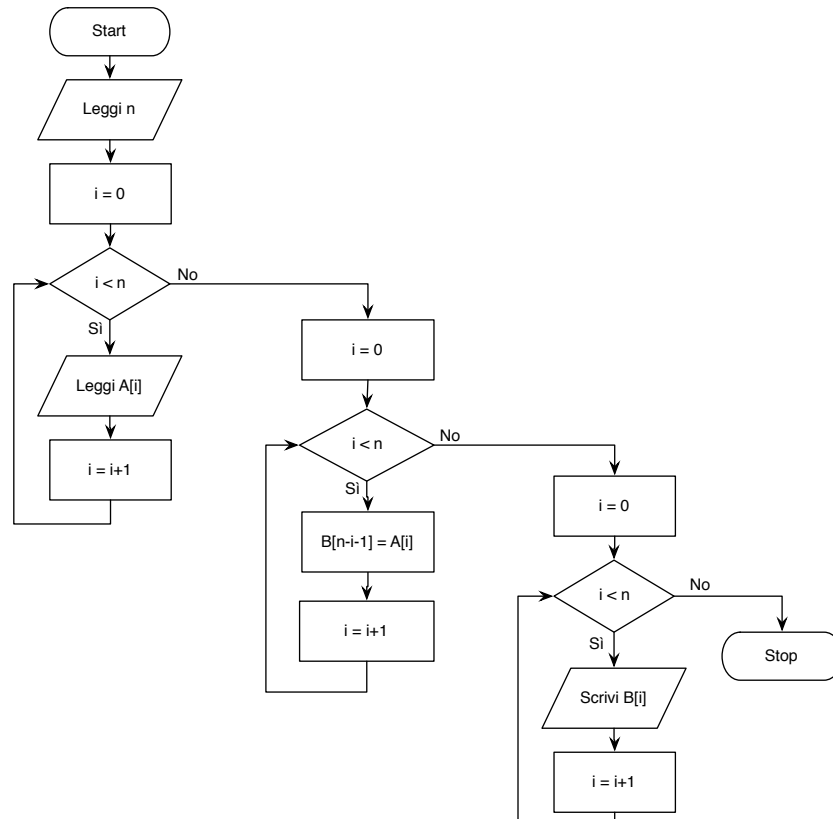
## Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 int main(void) {
6     int A[MAX], n, i, s;
7     float B[MAX];
8
9     printf("Numero di elementi: ");
10    scanf("%d", &n);
11    printf("Inserisci %d elementi: ", n);
12    for (i=0; i<n; i++)
13        scanf("%d", &A[i]);
14
15    s = 0;
16    for (i=0; i<n; i++) {
17        s = s + A[i];
18        B[i] = (float)s/(i+1);
19    }
20
21    printf("Vettore dei valori medi: ");
22    for (i=0; i<n; i++)
23        printf("%f ", B[i]);
24
25    return(0);
26 }
```

## Esercizio n. 2

Letto in input un array  $A$  di  $n$  interi, costruire un array  $B$  con gli stessi elementi di  $A$ , ma memorizzati al contrario (il primo elemento di  $A$  è l'ultimo elemento di  $B$ , l'ultimo elemento di  $A$  è il primo di  $B$ ). Stampare il vettore  $B$ .

### Diagramma di flusso



## Pseudo-codifica dell'algoritmo

- 1: leggi  $n$
- 2: leggi  $n$  numeri interi e memorizzali nell'array  $A$
- 3: **per**  $i = 0, 1, \dots, n - 1$  **ripeti**
- 4:    $B_{n-i-1} = A_i$
- 5: **fine-ciclo**
- 6: stampa l'array  $B$
- 7: stop

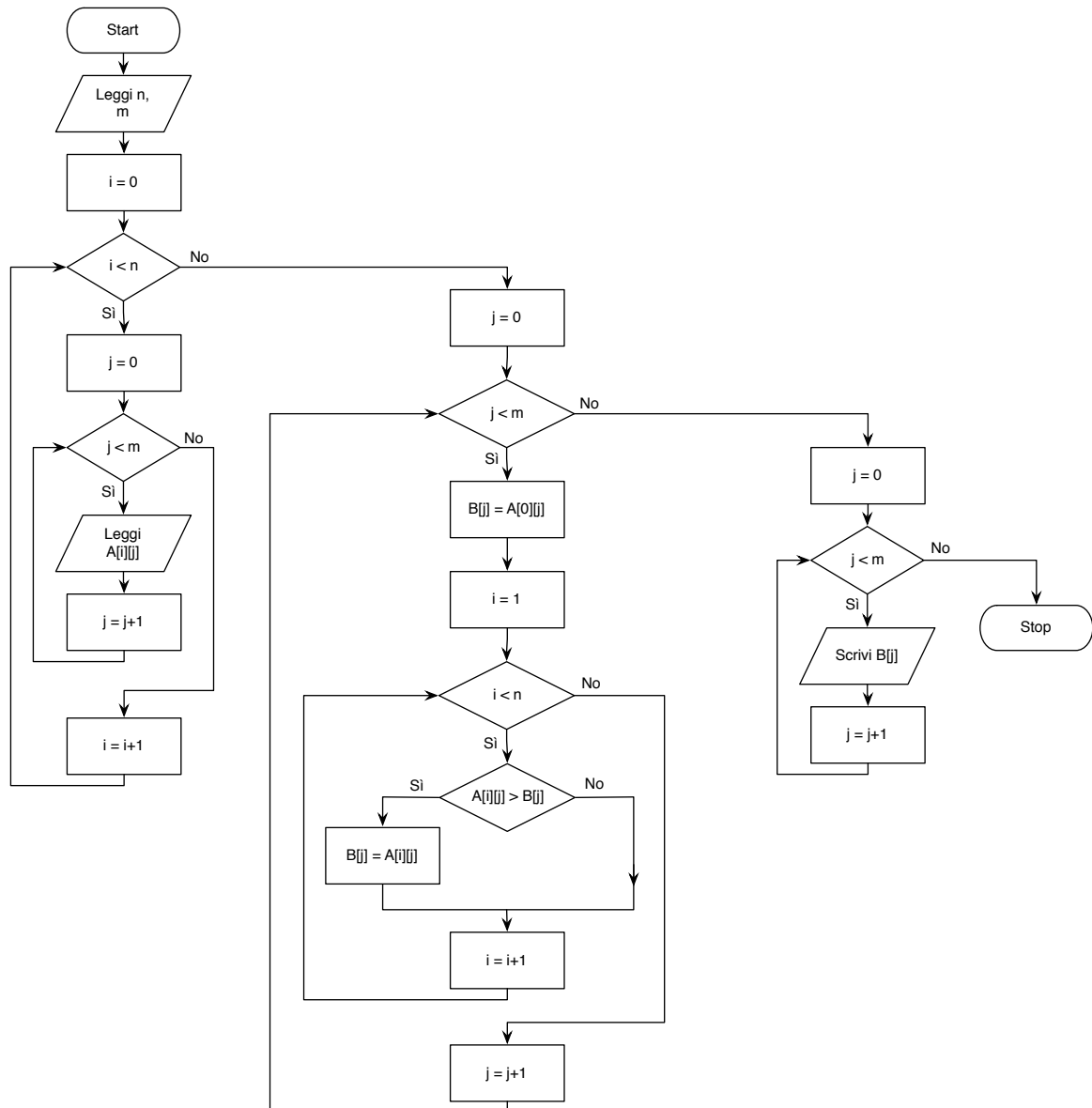
## Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 int main(void) {
6     int A[MAX], B[MAX], n, i;
7
8     printf("Numero di elementi: ");
9     scanf("%d", &n);
10    printf("Inserisci %d elementi: ", n);
11    for (i=0; i<n; i++)
12        scanf("%d", &A[i]);
13
14    for (i=0; i<n; i++)
15        B[n-i-1] = A[i];
16
17    for (i=0; i<n; i++)
18        printf("%d ", B[i]);
19
20    return(0);
21 }
```

### Esercizio n. 3

Letti in input due interi positivi  $n$  e  $m$ , leggere in input una matrice  $A$  di numeri interi, con  $n$  righe ed  $m$  colonne. Costruire un array  $B$  con  $m$  elementi e memorizzare in  $B_k$  ( $0 \leq k < m$ ) il massimo elemento della colonna  $k$  della matrice  $A$ . Stampare il vettore  $B$ .

#### Diagramma di flusso



## Pseudo-codifica dell'algoritmo

```
1: leggi  $n$  e  $m$ 
2: leggi in input gli elementi della matrice  $A$ 
3: per  $j = 0, 1, 2, \dots, m - 1$  ripeti
4:    $B_j = A_{0,j}$ 
5:   per  $i = 1, 2, \dots, n - 1$  ripeti
6:     se  $A_{i,j} > B_j$  allora
7:        $B_j = A_{i,j}$ 
8:     fine-condizione
9:   fine-ciclo
10: fine-ciclo
11: per  $j = 0, 1, 2, \dots, m - 1$  ripeti
12:   stampa  $B_j$ 
13: fine-ciclo
14: stop
```

## Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 int main(void) {
6     int A[MAX][MAX], B[MAX], n, m, i, j;
7
8     printf("Numero di righe e colonne: ");
9     scanf("%d %d", &n, &m);
10    for (i=0; i<n; i++) {
11        printf("Inserisci gli elementi della riga %d: ", i);
12        for (j=0; j<m; j++)
13            scanf("%d", &A[i][j]);
14    }
15
16    for (j=0; j<m; j++) {
17        B[j] = A[0][j];
18        for (i=1; i<n; i++) {
19            if (A[i][j] > B[j])
20                B[j] = A[i][j];
21        }
22    }
23
24    printf("Elementi massimi delle colonne della matrice:\n");
25    for (j=0; j<m; j++)
26        printf("%d ", B[j]);
27
28    return(0);
29 }
```