

# Corso di Algoritmi e Strutture Dati (IN110)

## Tutorato n. 5

Marco Liverani\*

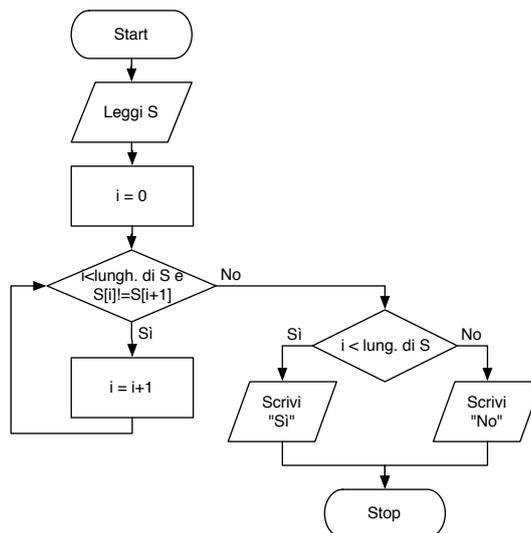
### Esercizio n. 1

Letta in input una stringa  $s$  stampare “Sì” se contiene lettere doppie, “No” altrimenti. Ad esempio se  $s = \text{“cassa”}$  allora stampa “Sì” perché ci sono le due “s”; se invece  $s = \text{“casa”}$  allora stampa “No”, anche se di lettere “a” ce ne sono due, ma non sono adiacenti.

#### Pseudo-codifica dell'algoritmo

- 1: leggi  $s$
- 2: per ogni carattere  $s_i$  ( $i = 0, 1, 2, \dots, n - 2$ ) verifica se è uguale al successivo ( $s_{i+1}$ )
- 3: se hai trovato almeno una coppia di caratteri consecutivi uguali, allora scrivi “Sì”, altrimenti scrivi “No”
- 4: stop

#### 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/>

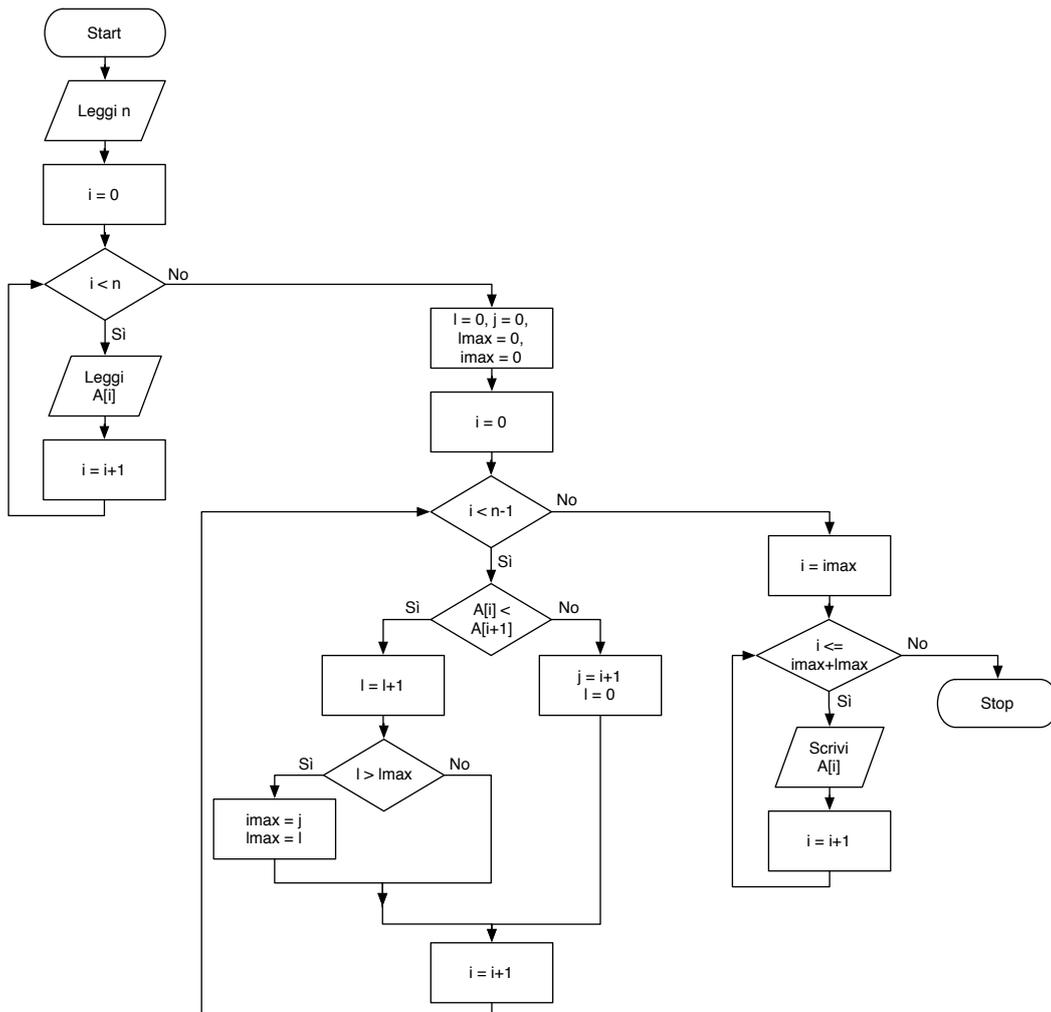
## Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <string.h>
4 #define MAX 30
5
6 int main(void) {
7     char s[MAX];
8     int i;
9     printf("Inserisci una parola: ");
10    scanf("%s", s);
11    for (i=0; i<strlen(s)-1 && s[i]!=s[i+1]; i++)
12        ;
13    if (i < strlen(s)-1)
14        printf("Si'\n");
15    else
16        printf("No\n");
17    return(0);
18 }
```

## Esercizio n. 2

Letta in input una sequenza di  $n$  numeri interi memorizzarla in un array  $A$ . Stampare la più lunga sottosequenza di numeri crescenti contigui in  $A$ .

### Diagramma di flusso



### Pseudo-codifica dell'algoritmo

- 1: leggi l'array  $A$  di  $n$  elementi
- 2: inizializza il contatore  $l = 0$  e poni  $l_{\max} = 0$
- 3: scorri l'array  $A$  dal primo al penultimo elemento (da  $A_0$  fino ad  $A_{n-2}$ ) ed esegui i passi 4-5:
- 4: se  $A_i < A_{i+1}$  allora incrementa il contatore  $l$
- 5: se  $l > l_{\max}$  allora memorizza la sequenza corrente come la sequenza più lunga
- 6: stampa la sequenza più lunga di lunghezza  $l_{\max}$
- 7: stop

### Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 30
4
5 int leggi_array(int A[]) {
6     int i, n;
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    return(n);
14 }
15
16 int main(void) {
17     int A[MAX], i, j, l, n, lmax, imax;
18     n = leggi_array(A);
19     l = 0;
20     j = 0;
21     lmax = 0;
22     imax = 0;
23     for (i=0; i<n-1; i++) {
24         if (A[i]<A[i+1]) {
25             l++;
26             if (l>lmax) {
27                 imax = j;
28                 lmax = l;
29             }
30         } else {
31             j = i+1;
32             l = 0;
33         }
34     }
35     for (i=imax; i<=imax+lmax; i++)
36         printf("%d ", A[i]);
37     printf("\n");
38     return(0);
39 }
```



### Pseudo-codifica dell'algoritmo

- 1: costruisci la matrice  $A$  di  $n$  righe ed  $m$  colonne
- 2:  $i_{\max} = 0, j_{\max} = 0, s_{\max} = 0$
- 3: **per**  $i = 0, 1, 2, \dots, n - 3$  **ripeti**
- 4:   **per**  $j = 0, 1, 2, \dots, m - 3$  **ripeti**
- 5:     costruisci la matrice  $3 \times 3$  che inizia in  $A_{i,j}$  e calcola la somma  $s$  dei suoi elementi
- 6:     **se**  $s > s_{\max}$  **allora**
- 7:       poni  $i_{\max} = i, j_{\max} = j, s_{\max} = s$
- 8:     **fine-condizione**
- 9:   **fine-ciclo**
- 10: **fine-ciclo**
- 11: stampa la matrice  $3 \times 3$  che inizia in  $A_{i_{\max}, j_{\max}}$
- 12: stop

### Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 50
5
6 void genera_matrice(int A[MAX][MAX], int *n, int *m) {
7     int i, j;
8     srand((unsigned)time(NULL));
9     printf("Inserisci il numero di righe e di colonne: ");
10    scanf("%d %d", n, m);
11    for (i=0; i<*n; i++)
12        for (j=0; j<*m; j++)
13        A[i][j] = rand() % 100;
14    return;
15 }
16
17 void stampa_matrice(int A[MAX][MAX], int n, int m) {
18     int i, j;
19     for (i=0; i<n; i++) {
20         for (j=0; j<m; j++)
21             printf("%4d", A[i][j]);
22         printf("\n");
23     }
24     return;
25 }
26
27 int somma_matrice(int A[MAX][MAX], int n, int m, int i, int j) {
28     int s, h, k;
29     s = 0;
30     for (h=0; h<3; h++)
31         for (k=0; k<3; k++)
32             s = s+A[i+h][j+k];
33     return(s);
34 }
35
36
37
```

```

38 int main(void) {
39     int A[MAX][MAX], n, m, somma, somma_max, i, j, i_max, j_max;
40     genera_matrice(A, &n, &m);
41     stampa_matrice(A, n, m);
42     somma_max = 0;
43     i_max = 0;
44     j_max = 0;
45     for (i=0; i<n-2; i++) {
46         for (j=0; j<m-2; j++) {
47             somma = somma_matrice(A, n, m, i, j);
48             if (somma > somma_max) {
49                 somma_max = somma;
50                 i_max = i;
51                 j_max = j;
52             }
53         }
54     }
55     for (i=i_max; i<i_max+3; i++) {
56         for (j=j_max; j<j_max+3; j++)
57             printf("%4d", A[i][j]);
58         printf("\n");
59     }
60     return(0);
61 }

```