

1) E' dato un programma che legge in input, rappresentato mediante una matrice di adiacenza, un grafo non orientato $G = (V;E)$ e assegna agli spigoli del grafo dei pesi interi non negativi. Legge in un array una sequenza S di vertici di G , verifica se il sottografo di G indotto da S è completo, e stampa il peso complessivo del sottografo di G indotto da S .

```
#include <stdlib.h>
#include <stdio.h>
#define MAX 100

int leggi_grafo(int M[MAX][MAX]) {
    int n, i, j;
    printf("Numero di vertici: ");
    scanf("%d", &n);
    for (i=0; i<n; i++) {
        M[i][i] = -1;
        for (j=i+1; j<n; j++) {
            printf("Peso dello spigolo (%d,%d), '-1'
                se lo spigolo non esiste: ", i, j);
            scanf("%d", &M[i][j]);
            M[j][i] = M[i][j]; } }
    return(n); }

int leggi_array(int a[]) {
    int n, i;
    printf("Numero di elementi: ");
    scanf("%d", &n);
    printf("Inserisci %d elementi: ", n);
    for (i=0; i<n; i++)
        scanf("%d", &a[i]);
    return(n); }

int main(void) {
    int G[MAX][MAX], S[MAX], i, j, flag=1,
        somma=0, n, m;
    n = leggi_grafo(G);
    m = leggi_array(S);
    for (i=0; i<m; i++)
        for (j=i+1; j<m; j++)
            if (G[S[i]][S[j]] == -1)
                flag = 0;
            else
                somma += G[S[i]][S[j]];
    if (flag)
        printf("Il sottografo indotto da S su G e'
                completo.\n");
    else
        printf("Il sottografo indotto da S su G
                NON e' completo.\n");
    printf("Il sottografo indotto da S su G pesa
            %d.\n", somma);
    return(1); }
```

a) Calcolare la complessità computazionale $T(n,m)$ dell'algoritmo in numero di passi base, con n e m dimensioni dell'input (*motivare brevemente le scelte e il metodo applicato durante il calcolo*);

b) Calcolare la complessità asintotica dell'algoritmo (*Procedere ri-calcolando la complessità di ogni macroistruzione, applicando opportunamente le regole per blocchi di istruzioni in sequenza o annidati.*)

2) Costruire una macchina di Turing che, data una sequenza di parentesi graffe e quadre in input, verifichi se la sequenza è bilanciata. Nel caso in cui la sequenza risulti bilanciata la macchina scriva sul nastro "S", altrimenti scriva "N". Si applichi la macchina al caso "[{} []" e al caso "[{ [] }]". Si descriva a parole, prima della matrice funzionale, l'algoritmo di cui la matrice sarà descrizione.