

ESERCIZI

1) Eseguire l'algoritmo TorriHanoi per $n = 5$, e scrivere la sequenza delle prime 13 mosse.

```
1 ----> 3
1 ----> 2
3 ----> 2
1 ----> 3
2 ----> 1
2 ----> 3
1 ----> 3
1 ----> 2
3 ----> 2
3 ----> 1
2 ----> 1
3 ----> 2
1 ----> 3
```

2) Dato un insieme di n elementi, scrivere un algoritmo GeneraCombinazioni per generare tutti i sottoinsiemi di k elementi.

Versione con vettore caratteristico

```
GeneraCombinazioni(A, k, b) // k <= b
if (k == 0) Elaborazione(A);
else if (k == b) {
    for (i = 0; i < b; i++) A[i] = 1;
    Elaborazione(A);
    for (i = 0; i < b; i++) A[i] = 0;
}
else { // k < b
    A[b-1]=1;
    GeneraCombinazioni(A, k-1, b-1);
    A[b-1]=0;
    GeneraCombinazioni(A, k, b-1);
}
```

Chiamata iniziale: GeneraCombinazioni(A, k, n), con $k \leq n$, con array A azzerato.

Complessità in tempo: $O(n^k)$

Versione con sottoinsieme

A contiene l'insieme di n elementi.

```
GeneraCombinazioni(A, C, k, m) // k <= m  
if (k == 0) Elabora(C);  
else if (k == m) {  
    for (i = 0; i < m; i++) C[i] = A[i];  
    Elabora(C);  
}  
else { // k < m  
    C[k-1]=A[m-1];  
    GeneraCombinazioni(A, C, k-1, m-1);  
    GeneraCombinazioni(A, C, k, m-1);  
}
```

Chiamata iniziale: Genera Combinazioni(A, C, k, n), con k <= n, e A vettore con contenente gli n elementi.

Complessità in tempo: $O(n^k)$

3) Scrivere un algoritmo di verifica per il problema di decidere se un grafo di n vertici, rappresentato con una matrice binaria M, contiene una clique di k vertici, e dimostrare che è polinomiale.

M: matrice binaria n x n che rappresenta il grafo

C: certificato polinomiale costituito da un array contenente il sottoinsieme di k vertici da verificare

```
VerificakClique(M, C)  
for (i = 0; i < k; i++) {  
    for (j = i+1; j < k; j++) {  
        if (M[C[i]][C[j]] == 0) return false;  
    }  
}  
return true;
```

Costo in tempo: $O(k^2)$

4) Scrivere un algoritmo per trovare una clique di k vertici in un grafo di n vertici.

A: insieme dei vertici del grafo

M: matrice di adiacenza ($n \times n$)

```
TrovakClique(M, A, C, k, m) // k <= m
if (k == 0) {
    if (VerificakClique(M,C))
        * stampa la clique rappresentata da C e termina;
}
else if (k == m) {
    for (i = 0; i < m; i++) C[i] = A[i];
    if (VerificakClique(M,C) == true)
        * stampa la clique rappresentata da C e termina;
}
else { // k < m
    C[k-1]=A[m-1];
    TrovakClique(M, A, C, k-1, m-1);
    TrovakClique(M, A, C, k, m-1);
}
```

Complessità al caso peggio: $O(n^k k^2)$

5) Completare l'esercizio di pag 19 ([CGGR], prologo).

A: insieme di n elementi positivi

somma: somma cercata

b: indice

```
SottoinsiemeDiSommaData(A, somma, b) {
if (b > 0) {
    if (somma == A[b-1]) * stampa trovato sottoinsieme e termina
    else if (somma > A[b-1])
        SottoinsiemeDiSommaData (A, somma - A[b-1], b-1);
        SottoinsiemeDiSommaData (A, somma, b-1);
}
}
```