

Esercizio 1

Si analizzi la complessità del seguente programma C++, considerando sia le condizioni di caso migliore che le condizioni di caso peggiore.

```
bool trovato(int x,int V[], int inf, int sup)
{ for (int i=inf; i<=sup; i++)
    if (V[i]==x)
        return true;
    return false;
}

void cancella (int V[], int pos, int& dim)
{
    for (int i=pos; i<(dim-1); i++)
        V[i]=V[i+1];
    dim--;
}

void compatta (int V[], int & dim)
{ int i=1;
  while (i<dim)
  {
    if (trovato(V[i],V,0,i-1))
        cancella(V,i,dim);
    else
        i++;
  }
}

void main()
{
    int V[N];
    compatta(V,N);
}
```

Esercizio 2

Studiare la complessità nel caso migliore e nel caso peggiore della seguente funzione,

```
void elabora(int M[][N]) {
    bool b=false;
    int i = 0, k = 0, j = N;
(1) while (i<N/2 && j>N/2 && !b) {
(2)     if ( f1(M, k++) )
        b = true;
(3)     if ( f2(M) ) {
        i = max( 0, i-1 );
        j = max( N/2, j-1 );
    }
    else {
        i = min( i+1, N/2 );
        j = min( j+1, N );
    }
}
}
```

sapendo che la funzione **f1(M,k)** ha la seguente complessità:

per $k < N$ $O(N)$

e può terminare restituendo indifferentemente True o False

per $k \geq N$ $O(1)$

e termina restituendo sempre True

mentre la funzione **f2(M)** ha la seguente complessità:

$O(N^2)$

e può terminare restituendo indifferentemente True o False

N.B. le funzioni min(...) e max(...) si possono considerare con complessità costante $O(1)$.