CASO MEDIO INSERTION SORT

$$T(n) = 8n - 6 + 4\left(2\sum_{2}^{n} t_j - (n-1)\right)$$

Ricordiamo che $\forall\, 2 \le j \le n \quad 1 \le t_j \le j$

Per la media aritmetica avremo (somma di tutti i valori di t_j fratto j):

$$\frac{\sum_{1}^{j} k}{j} = \frac{\cancel{j}(j+1)}{2\cancel{j}} = \frac{j+1}{2} \text{ ripetizioni medie}$$

Quindi il tempo medio di esecuzione è:

$$T_{MD}(n) = 8n - 6 + 4\left(\cancel{2}\sum_{2}^{n} \frac{j}{\cancel{2}} - (n-1)\right) = 8n - 6 + 4\sum_{2}^{n} j - 4n + 4 =$$

$$= 4n - 2 + 4\left(\frac{n(n+1)}{2}\right) = 2n^2 + 6n - 6 = \Theta(n^2)$$

Quindi anche nel caso medio abbiamo un tempo quadratico.

ALGORITMI RICORSIVI

Vediamo il calcolo di un fattoriale

$$n! \begin{cases} 1 & n = 1 \\ n(n-1)! & n > 1 \end{cases}$$

Il cui algoritmo sarà

FATT (N)

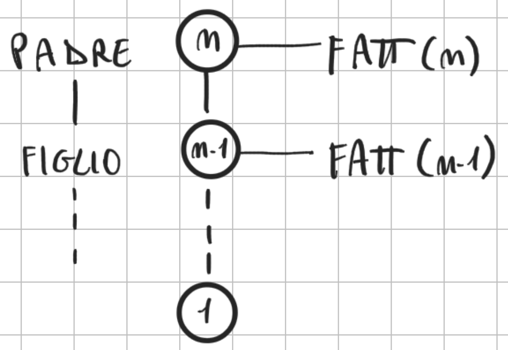
IF N > 1 THEN	$\Theta(1)$
X = FATT (N-1)	?
RIS = X · N	$\Theta(1)$
ELSE	
RIS = 1	$\Theta(1)$
RETURN RIS	$\Theta(1)$

Scriviamo in maniera ricorsiva la funzione di tempo:

$$T(n) = \begin{cases} \Theta(1) & n = 1 \\ \Theta 1 + T(n-1) & n > 1 \end{cases}$$

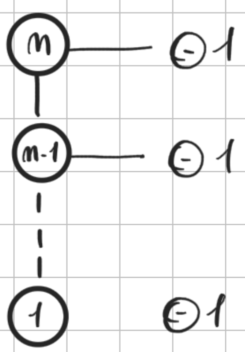
Vogliamo trovare la forma chiusa di T(n) così potremo trovare il valore esatto della nostra equazione di secondo ordine.
In particolare dobbiamo capire quanto sia lunga la sequenza generata dalla ricorsione, quindi il numero esatto di operazioni.

La relazione tra le chiamate ricorsive padre-figlio è rappresentata da un ALBERO.



ALBERO DEGENERE
(1 FIGLIO PER NODO)

Il tempo totale è dato dalla somma di tutte le chiamate (tutti i nodi dell'albero). Ogni chiamata costa $\Theta 1$ + la chiamata successiva che costerà $\Theta 1$ e così via fino al caso base (FOGLIA)



Il livello i di una foglia è tale che $n-i=1$ perché rappresentano il caso base (input 1) quindi $i=n-1$.

Il tempo totale sarà la somma del costo di ogni chiamata i volte (livello delle foglie = numero di chiamate totali)

Quindi $\sum_{0}^{n-1} \Theta 1 = \Theta n \Rightarrow T(n) = \sum_{0}^{n-1} \Theta(1) = \Theta n$

MERGE SORT

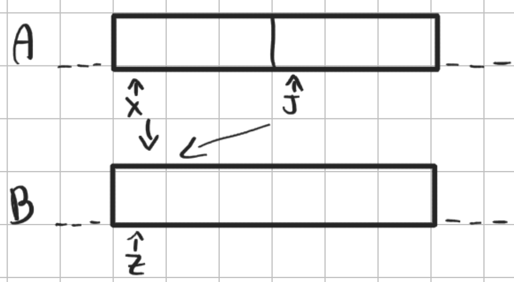
L'algoritmo di merge sort si basa sul concetto di DIVIDE ET IMPERA ovvero scomporre un problema in sottoproblemi più elementari della stessa natura.

Quindi l'idea è di prendere una sequenza, dividerla e ordinare le sequenze in modo ricorsivo (dividendo ancora fino ad arrivare a un caso base elementare)

```
MERGE_SORT (A, i, j)                      i e j sono gli indici di inizio e fine
                                          sequenza da ordinare
    IF i < j THEN        }
                         } Θ1
        q = (i+j)/2      }                DIVIDI LA SEQUENZA

        MERGE_SORT (A, i, q)              ORDINA LA PRIMA METÀ

        MERGE_SORT (A, q+1, j)            ORDINA LA SECONDA METÀ

        MERGE (A, i, q, j)      Θn        UNISCI LE SEQUENZE ORDINATE
```

$$T(n) = \begin{cases} \Theta 1 & n \leq 1 \quad (\text{perché } n = j - i + 1) \\ \Theta n + \underbrace{T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right)}_{2T\left(\frac{n}{2}\right)} & n > 1 \end{cases}$$

```
MERGE (A, i, q, j)

x = i

y = q+1

z = i

WHILE (x ≤ Q AND y ≤ J) DO

    IF A[x] < A[y]

        B[z] = A[x]

        x++

    ELSE

        B[z] = A[y]

        y++

    z++

WHILE (x ≤ Q) DO

    B[z] = A[x]

    x++

    z++

WHILE (x ≤ y) DO

    B[z] = A[y]

    z++

    y++
```



```
FOR z = i TO j DO
    A[z] = B[z]
RETURN A[z]
```