MergeSort

- Si descriva a parole, in non più di 10 righe, il funzionamento dell'algoritmo MergeSort, **oppure** se ne scriva il codice omettendo quello della funzione MERGE che può essere descritta a parole in non più di 4 righe.
- Si scriva di conseguenza la relazione di ricorrenza che descrive la complessità in tempo di MergeSort.

* Se ne indichi la soluzione, motivando la risposta usando uno qualsiasi dei metodi visti a lezione.

L'algoritmo MergeSort è un algoritmo di ordinamento ricorsivo che utilizza la tecnica del *divide et impera* che consiste nella suddivisione del problema in sottoproblemi di dimensione sempre più piccola.
Funziona in questo modo:
Se la sequenza da ordinare ha lunghezza 0 oppure 1, è già ordinata. Altrimenti, prima la sequenza viene divisa in due metà, successivamente viene ordinata la prima parte e in seguito la seconda applicando ricorsivamente l'algoritmo, e infine le due parti ordinate vengono fuse (*merge*). Per fare questo, si estrae ripetutamente il minimo delle sottosequenze e lo si pone nel vettore in uscita.

oppure

```
void mergeRicorsivo(int A[ ], int from, int to)
{
int mid;
if (from < to) {            /* l'intervallo da mid a to, estremi inclusi,
                               comprende almeno due elementi */
        mid = (from + to) / 2;
        mergeRicorsivo(A, from, mid);
        mergeRicorsivo(A, mid+1, to);
        merge(A, from, mid, to);            /* fonde le due porzioni ordinate
                                               [from, mid], [mid+1, to] nel
                                                  sottovettore [from, to]  */
} }
```

La funzione Merge, fonde le due porzioni dell'array A con indici compresi tra from e mid e tra mid+1 e to. La procedura utilizza un array di supporto B. Merge copia in modo ordinato gli elementi delle due parti di A in B fino a esaurire una delle due, infine ricopia tutto da B in A.

La procedura Merge ha costo $O(n)$ perché ogni elemento viene confrontato con gli altri e viene spostato o meno, tuttavia viene spostato nella sua posizione definitiva al più una volta. Il MergeSort, chiama ricorsivamente sé stesso e usa Merge per unire i risultati: MergeSort richiama sé stessa due volte, e ogni volta su metà della sequenza in input (n/2), la complessità al caso pessimo è quindi $\Theta(n \log n)$.