

Merge sort

Complessità di merge: $\Theta(n)$.

Complessità di mergesort:

$$T(n) = \begin{cases} \Theta(1) & n \leq 1 \\ 2T(\frac{n}{2}) + \Theta(n) & n \geq 2 \end{cases} = \Theta(n \log n).$$

Non ordina in loco, complessità in spazio $\Theta(n)$.

```
def mergesort(A, p, r):  
    if p < r:  
        q = int((p + r) / 2)  
        mergesort(A, p, q)  
        mergesort(A, q + 1, r)  
        merge(A, p, q, r)
```

```
def merge(A, p, q, r)  
    n = q - p + 1  
    m = r - q  
  
    L = []  
    R = []  
    L[n] = R[m] = +inf.0  
  
    for i in range(n):  
        L[i] = A[p + i]  
    for j in range(m):  
        R[j] = A[q + j + 1]  
  
    i = j = 1  
    for k in range(p, r + 1):  
        if L[i] <= R[j]:  
            A[k] = L[i]  
            i += 1  
        else:  
            A[k] = R[j]  
            j += 1
```