-Il Mergesort è un algoritmo di ordinamento ricorsivo che si basa sulla divisione di un array in parti il più piccole possibile, ordinare singolarmente queste ultime e successivamente unirle (merge) in un unico array ordinato. Il processo di divisione dell’array finisce quando quest’ultimo risulta avere meno di due elementi. L’algoritmo per eseguire il Mergesort è necessariamente ricorsivo e necessita di un vettore temporaneo in cui immettere i parziali array ordinati, quest’ultimo causa un peso maggiore dell’algoritmo, che risulta essere tuttavia più veloce di altri algoritmi occupanti meno memoria.

Schema di funzionamento del Mergesort

-TEMPI DI ESECUZIONE

TEMPO NEL CASO PEGGIORE: **O(n log (n))**

TEMPO NEL CASO MIGLIORE: nel caso migliore e medio il tempo di esecuzione rimane lo stesso del caso peggiore in quanto tutti i controlli e le operazioni vengono eseguiti nello stesso numero definito di volte.

SPAZIO UTILIZZATO: lo spazio utilizzato dal merge sort è necessariamente maggiore rispetto a quello utilizzato da altri algoritmi di ordinamento comuni dato che vi è la necessita di utilizzare un array di temporaneo per l’ordinamento degli elementi

**PSEUDOCODICE:**

**Funzione Sort**(Vettore A (A1, ...., An), numero iniziale (nIniz) , numero finale (nFin))

Intero middle

Se (nFin > nIniz) allora

calcola la metà (middle)

Sort(A, nIniz, middle)

Sort(A, middle + 1, nFin)

Merge(A, nIniz, middle + 1, nFin)

Altrimenti non fa nulla, non è divisibile ulteriormente

**Funzione Merge**(A, nIniz, middle, nFin)

Intero i,primo numero a sinistra della meta del vettore(leftend),primonumero del vettore(left),primo numero a destra della metà (rightstart)

Input Vettore di copia del vettore A(v2)

finchè (left <= leftend E middle <= right)

se( A[left] <= A[middle] )

copia il primo numero del vettore A nel vettore v2

altrimenti

copia il middle del primo vettore in v2

finchè (left <= leftend)

copia i numeri restanti della sezionedi sinistra di A in v2

finche(middle <= right)

copia i numeri restanti della sezione di destra di A in v2

per (i=0; i < rightstart; i++)

copia il vettore di copia(v2) nel vettore originale(A)