**Algoritmo Merge Sort**

**pseudocodigo:**

*# Partido a la mitad*

m = n / 2

*# orden recuersivo*

sort a[1..m]

sort a[m+1..n]

*# fusionar sub-arrays ordenados usando arreglo temporal*

b = copiar a[1..m]

i = 1, j = m+1, k = 1

mientras i <= m and j <= n,

a[k++] = (a[j] < b[i]) ? a[j++] : b[i++]

→ *invariante: a[1..k] en la posicion final*

mientras i <= m,

a[k++] = b[i++]

→ *invariante: a[1..k] en la posicion final*

**Propiedades**

Estable

Θ (n) más espacio para las matrices (como se muestra)

Θ (lg (n)) espacio extra para listas enlazadas

Θ (n · lg (n)) tiempo

No adaptativo

No requiere acceso aleatorio a los datos

**Discusión**

Ordenamiento por mezcla es muy predecible. Se hace entre 0,5 \* lg (n) y lg (n) comparaciones por elemento, y entre lg (n) y 1.5 \* lg (n) intercambia por elemento. Los mínimos se consiguen los datos ya ordenados; los máximos se alcanzan, en promedio, para los datos aleatorios. Si utiliza Θ (n) más espacio no es de preocupación, y luego fusionar especie es una excelente opción: Es fácil de implementar, y es la única estable O (n · lg (n)) algoritmo de clasificación. Debemos tener en cuenta que cuando se ordenan las listas enlazadas, fusionar especie requiere sólo Θ (lg (n)) espacio adicional (por recursividad).

Ordenamiento por mezcla es el algoritmo de elección para una variedad de situaciones: cuando se requiere la estabilidad, al ordenar listas enlazadas, y cuando el acceso aleatorio es mucho más caro que el acceso secuencial (por ejemplo, la clasificación externa en la cinta).

Sí existen el tiempo lineal en el lugar donde se funden los algoritmos para el último paso del algoritmo, pero son caros y complejos. La complejidad se justifica para aplicaciones tales como la clasificación externa cuando Θ (n) más espacio no está disponible.