

# Actividad 2.3

---

## Reflexión

---

16 de junio 2022

Félix Javier Rojas Gallardo

A01201946

Con el manejo de memoria dinámico, podemos hacer que los objetos que se pasen por el merge sort ocupen mucho menor espacio en la memoria.

## Complejidad temporal

---

Como el mergesort en su peor caso es  $O(n \log n)$ , es mejor que el peor caso de quicksort  $O(n^2)$ , particularmente en este tipo de estructuras de datos pues sólo se pueden recorrer linealmente.

Dado que quicksort también utiliza acceso aleatorio para evitar sus peores casos, las listas ligadas simples y doblemente ligadas no permiten estos accesos aleatorios, por lo que las optimizaciones del quick sort no son posibles aquí.

## Complejidad espacial

---

Justamente como utilizamos los mismos nodos para recorrer el arreglo, el espacio utilizado pasa de ser  $O(n)$  a  $O(1)$  pues utilizamos nodos para intercambiar la información, en vez de generar un array para guardar todos los elementos.

## DLL vs SLL

De haber utilizado una lista ligada simple hubiésemos necesitado implementar una pila o fila para almacenar los datos que estamos moviendo, lo que incrementaría la complejidad espacial de nuestro programa. Esto sería un problema mayor también si nuestra bitácora tuviese un tamaño mucho mayor. La complejidad de ese espacio ocupado a procesar, que ya por sí tiene múltiples entradas sería tremendo, aunque sí poseemos el poder computacional para realizarlo.