Jaime Eduardo López Castro - A00833173

Reflexión:

En esta implementación utilizamos una lista doblemente ligada en vez de una sencilla, esto tiene la ventaja de que podemos recorrer el contenido de nuestra bitácora al revés en vez con complejidad lineal en vez de cuadrática, como sería con una lista simple. La única desventaja que presenta esto es que una lista doblemente enlazada utiliza más memoria ya que cada nodo tiene como atributo también un apuntador a su nodo previo.

La complejidad final de nuestra implementación es de O(log n) para la búsqueda de un elemento después de ordenar la bitácora ya que utilizamos el algoritmos de búsqueda binaria y O(n log n) para ordenar la bitácora utilizando nuestra implementación de MergeSort. Las complejidades de la lectura e impresión son lineales, ya que tenemos que visitar todos los nodos existentes en nuestra lista.

Siguiendo con la comparación entre una lista ligada y una lista doblemente ligada; las complejidades en sus operaciones básicas es la misma: O(1) para insertar y eliminar elementos al principio y al final gracias a los apuntadores de head y tail, O(logn) para buscar un elemento con la bitácora ordenada, como ya había mencionado y O(n) para buscarlo en la lista desordenada ya que estas no funcionan por indexado como los vectores o los arrays; O(n) para insertar o eliminar elementos a puntos intermedios de la lista, ya que tenemos que recorrerla toda elemento por elemento para poder llegar a la posición deseada. En nuestro caso no vamos a insertar ni borrar elementos entre medio de la lista, por lo que sólamente nos interesa saber su complejidad para insertar, borrar y ordenar la bitácora.

En cuanto al algoritmo de ordenamiento que utilizamos, implementamos los 2 y los comparamos en tiempo promediando 5 tests para así poder saber cuál resulta mejor en nuestro caso.

Jaime Eduardo López Castro - A00833173

QuickSort: promedio user - 0.214 ms

```
time ./main

--- Creando una lista doblemente ligada vacia --- 0x7ffd821c2060
--- Liberando memoria de la lista doblemente ligada --- 0x7ffd821c2060

real    0m0.337s
user    0m0.211s
sys 0m0.050s
time ./main

--- Creando una lista doblemente ligada vacia --- 0x7ffc08e76300
--- Liberando memoria de la lista doblemente ligada --- 0x7ffc08e76300

real    0m0.373s
user    0m0.196s
sys 0m0.052s
time ./main

--- Creando una lista doblemente ligada vacia --- 0x7ffd37af6a20
--- Liberando memoria de la lista doblemente ligada --- 0x7ffd37af6a20

real    0m0.300s
user    0m0.225s
sys 0m0.031s
time ./main

--- Creando una lista doblemente ligada vacia --- 0x7ffe2dc7e410
^[[A--- Liberando memoria de la lista doblemente ligada --- 0x7ffe2dc7e41

real    0m0.956s
user    0m0.253s
sys 0m0.049s
time ./main

--- Creando una lista doblemente ligada vacia --- 0x7ffd61386150
--- Liberando memoria de la lista doblemente ligada --- 0x7ffd61386150

real    0m0.575s
user    0m0.204s
sys 0m0.050s
time ./main

--- Creando una lista doblemente ligada vacia --- 0x7ffd2fe735a0
--- Liberando memoria de la lista doblemente ligada --- 0x7ffd2fe735a0

real    0m0.323s
user    0m0.195s
sys 0m0.061s
```

Jaime Eduardo López Castro - A00833173

MergeSort: promedio user - 0.0702 ms

```
time ./main

--- Creando una lista doblemente ligada vacia --- 0x7ffc0957d8b0
--- Liberando memoria de la lista doblemente ligada --- 0x7ffc0957d8b0

real    0m0.304s
user    0m0.089s
sys 0m0.040s
time ./main

--- Creando una lista doblemente ligada vacia --- 0x7fffd63d6220
--- Liberando memoria de la lista doblemente ligada --- 0x7fffd63d6220

real    0m0.145s
user    0m0.050s
sys 0m0.076s
time ./main

--- Creando una lista doblemente ligada vacia --- 0x7fffea4bcbc0
--- Liberando memoria de la lista doblemente ligada --- 0x7fffea4bcbc0

real    0m0.112s
user    0m0.060s
sys 0m0.045s
time ./main

--- Creando una lista doblemente ligada vacia --- 0x7ffe474c9fe0
--- Liberando memoria de la lista doblemente ligada --- 0x7ffe474c9fe0

real    0m0.138s
user    0m0.060s
sys 0m0.056s
time ./main

--- Creando una lista doblemente ligada vacia --- 0x7fff64d32f30
--- Liberando memoria de la lista doblemente ligada --- 0x7fff64d32f30

real    0m0.156s
user    0m0.092s
sys 0m0.036s
```

Jaime Eduardo López Castro - A00833173

Fuentes:

- GeeksforGeeks. (2020, 1 diciembre). Advantages, Disadvantages, and uses of Doubly Linked List. https://www.geeksforgeeks.org/advantages-disadvantages-and-uses-of-doubly-linked-list/
- GeeksforGeeks. (2022, 3 julio). Merge Sort for Doubly Linked List. https://www.geeksforgeeks.org/merge-sort-for-doubly-linked-list/
- GeeksforGeeks. (2022a, junio 24). Binary Search on Singly Linked List. https://www.geeksforgeeks.org/binary-search-on-singly-linked-list/