



FCEFyN

Facultad de
Ciencias Exactas
Físicas y Naturales

Algoritmos y Estructuras de Datos

Trabajo Práctico N° 2

- **Integrantes**

Robledo Valentín (42555866)

Bottini Franco Nicolas (41525391)

- **Carrera**

Ingeniería en Computación.

- **Consigna**

La tarea consiste en ordenar alfabéticamente un archivo de texto con palabras, las cuales algunas pueden estar repetidas y otras no, generando otro archivo de texto con dichas palabras ya ordenadas.

- **Año**

2021



Desarrollo

Comenzamos desarrollando una clase “Tree” la cual se encarga de generar un árbol n-ario que será relleno de palabras, y además cumple la propiedad de ser un árbol semicompleto (cada nodo deberá tener n hijos a excepción de las hojas).

Además de esto, una misma palabra no generará más de un nodo dentro del árbol, sin embargo, se llevará la cuenta de las repeticiones en el nodo que corresponda.

Luego de esto, utilizando como clase base a “Tree” implementamos la clase “Heap” donde los elementos insertados en el mismo se ordenan de tal manera que un nodo padre siempre sea mayor a todos sus nodos hijos.

Finalmente, en esta clase implementamos el método “sort” que se encarga de ordenar las instancias de la clase “Heap”.

Dentro de la función “main” nos encargamos de leer el archivo “Input.txt” e ingresar las palabras dentro de una instancia de la clase “Heap”, reemplazando las vocales acentuadas por sus equivalentes sin acentuación, la letra “ñ” por la “n”, etc. y eliminando caracteres tales como: “,” - “.” - “()” - “¿?”, entre otros. Posterior a esto y tras ordenar el respecto heap generamos el archivo “Output.txt” donde figura la lista de palabras ingresadas ordenadas alfabéticamente con sus correspondientes repeticiones y el número de comparaciones realizadas.

Análisis

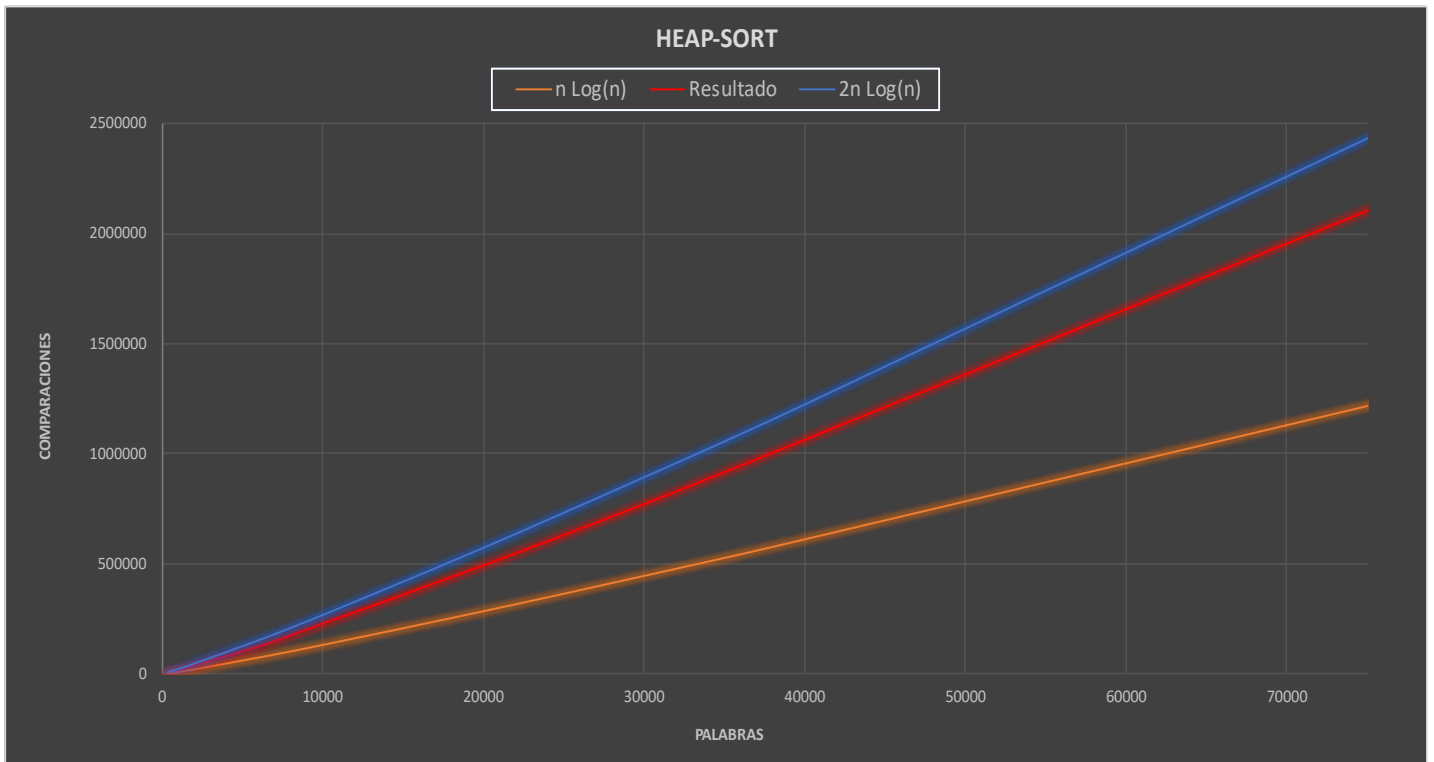
Realizando una serie de pruebas utilizando un heap binario, comprobamos que la complejidad del algoritmo efectivamente nos da en el orden de $O(n \log(n))$.

Obteniendo los siguientes resultados:

Tabla:

Palabras	$n \log(n)$	Resultado	$2n \log(n)$
333	2790	3979	5581
1535	16246	24978	32493
6269	79077	127366	158155
6860	87424	141112	174848
9917	131655	223459	263310
19781	282311	484440	564622
29699	441272	761194	882543
39707	606608	1053909	1213216
79356	1291602	2234880	2583205

Gráfico:





FCEPyN

Facultad de
Ciencias Exactas
Físicas y Naturales

Por otro lado, manteniendo fijo el número de palabras y realizando el ordenamiento con heaps de distinto orden obtuvimos que el número de comparaciones aumenta de manera aproximadamente lineal con el incremento del orden del heap.

Adjuntamos gráfico:

