Alejandro Lizárraga Vizcarra A01284610

Reflexión Evidencia 3

Para esta evidencia diseñó un programa que lee información de un archivo. Esta información decidimos que se guardaría en diferentes vectores de C++. Se almacenarían de tal modo que hay un almacenamiento de los datos con entrada del mar mediterráneo, mar rojo y otro con la mezcla de ambos. Una vez que tenemos esta información almacenada, podemos utilizar diferentes procedimientos para administrar la información.

Se utilizó el HeapSort, al pasar el vector del mar requerido a una estructura de datos diseñada para utilizar métodos tipo Heap. Este ordenamiento tiene un BigO en el peor caso de O(n log(n)). Es similar a un binary tree pero con propiedades Heap con el número más grande, se pudo utilizar para ordenamiento.

Por el otro lado se utilizó un árbol binario para la búsqueda según un identificador llamado ubi. A este árbol se le introdujeron los datos de manera secuencial según el documento, y la estructura de datos que manejamos agregaba al árbol los datos de manera organizada. Debido a esto, los datos están guardados de una manera no organizada. Con esto podemos utilizar el método de búsqueda binaria del árbol binario. Es importante meter los datos de una manera no ya *sorted* porque si no el uso de un árbol binario sería igual a una lista enlazada común. Según esto tenemos una complejidad de O(log(n).

El uso de estas estructuras de datos llevan a soluciones más rápidas para los problemas que se presentaron, aunque es importante mencionar que debido a nuestra solución con la creación de múltiples vectores y estructuras diferentes, aunque la complejidad puede ser buena y la solución rápida, el programa pudiera llegar a tener un uso de memoria pesado.