

Act 3.4 - Reflexión Actividad Integral de BST

Para esta actividad utilicé el ordenamiento por Heap Sort es un algoritmo de ordenamiento no recursivo, no estable, con complejidad computacional $\Theta(n \log n)$, el cual es uno de los más eficientes que hemos visto según yo. Este algoritmo consiste en almacenar todos los elementos del vector a ordenar en un montículo (heap), y luego extraer el nodo que queda como nodo raíz del montículo (cima) en sucesivas iteraciones obteniendo el conjunto ordenado. (colaboradores Wikipedia, 2021).

También cabe destacar el Max Heap que también utilice en esta actividad, el cual es una estructura de datos especializada basada en árboles que es esencialmente un árbol casi completo que satisface la propiedad del montón : en un montón máximo , para cualquier nodo C dado , si P es un nodo padre de C, entonces la clave (el valor) de P es mayor o igual que la clave de C. (colaboradores Wikipedia, 2021).

También fue un poco complicado realizar esta actividad porque tuve que hacer la nueva clase de lp para poder leerlo de la manera correcta y ordenarlo por lp, fue un proceso largo y la verdad sí tuve muchas complicaciones y errores al momento de hacerlo. Lo bueno es que tenemos herramientas como Geeks for Geeks donde pude tomar el algoritmo de Heap e implementarlo en mi programa, adaptándolo al mismo; también que la base del programa pasado también se pudo volver a usar en esta actividad y no tuve que volver a empezarla de cero. También me facilitó hacerla con vectores ya que se me hacen más familiares y más fáciles de utilizar que la double linked list, la cual la actividad pasada si me costó mucho.

Al final pude resolver el último paso con un contador el que da como resultado el número de lps repetidas y pues ya las imprimí en el archivo con las 5 lps más repetidas, la lp y el número de veces o accesos totales que hubo de cada una.

Referencias:

GeeksforGeeks. (2021, 15 septiembre). HeapSort. Recuperado de

<https://www.geeksforgeeks.org/heap-sort/>

colaboradores de Wikipedia. (2021a, febrero 16). Heapsort. Recuperado de

<https://es.wikipedia.org/wiki/Heapsort>

Wikipedia contributors. (2021b, noviembre 5). Heap (data structure). Recuperado de

[https://en.wikipedia.org/wiki/Heap_\(data_structure\)](https://en.wikipedia.org/wiki/Heap_(data_structure))