Instituto Tecnológico y de Estudios Superiores de Monterrey

Programación de estructuras de datos y algoritmos fundamentales (Gpo 602)

Reflexión Evidencia 3

Nombre de los profesores
David Alonso Cantú Delgado

Hecho por:
Hugo Ochoa López Portillo A00835999

Sábado 28 de octubre del 2023

Estructuras de datos jerárquicas

Una estructura de datos jerárquica es una estructura en la cual la información es almacenada jerárquicamente al acomodar los registros en forma de árbol. Estos tienen un nodo el cual contiene la información y además apuntadores a su padre y posibles hijos. El objetivo de organizar la información así es tener un programa más eficiente, ya que no se tiene que recorrer uno por uno algo como una fila o una pila.

En clases se vieron las siguientes estructuras de datos jerárquicas:

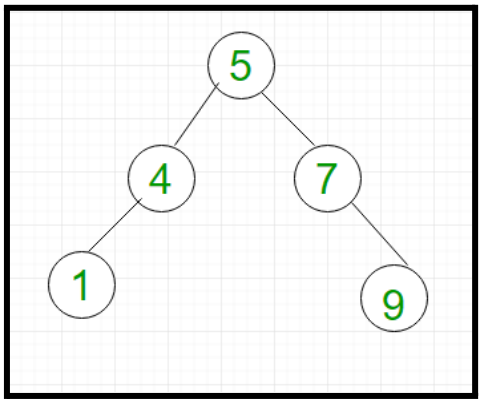
- Árboles binarios
- AVL
- HeapSort

Para esta evidencia se usaron los árboles binarios y heap sort.

Árboles binarios

En esta estructura de datos, se tiene una forma de árbol, se empieza por la raíz la cual está sola y esta puede tener dos hijos, uno a la izquierda y uno a la derecha, la condición es si el valor del dato hijo es menor al padre entonces se acomoda la izquierda, si es igual o mayor al valor del padre se acomoda a la derecha. Es una estructura jerárquica intuitiva, lo malo es que se puede desbalancear fácilmente hacia un lado y se vuelve lenta.

En el peor de los casos los árboles binarios tienen la complejidad de O(n), donde n es la altura del árbol. (GeeksforGeeks, 2022)



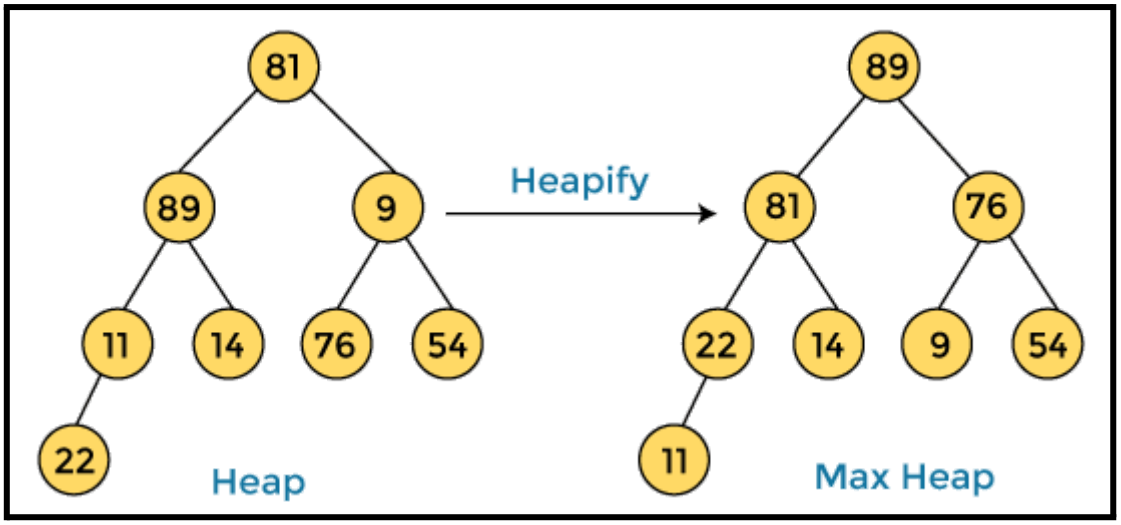
Ejemplo de árbol binario de búsqueda

HeapSort

El HeapSort también tiene forma de árbol, primero se realiza un Heap o un MinHeap que es un árbol donde igual se tienen dos hijos y se busca llenar nivel por nivel, dándole preferencia a la izquierda. En estas estructuras la raíz es el dato con mayor prioridad, en un Heap es el dato con mayor valor y en un MinHeap es el dato con menor valor.

El HeapSort consiste en quitar siempre el mayor elemento el cual siempre va a ser la raíz del Heap, una vez seleccionado el máximo, lo intercambiamos con el último elemento del vector, disminuimos la cantidad de elementos del Heap y nos encargamos de acomodarlo para que vuelva a ser un Heap.

Esta estructura de datos jerárquica tiene una complejidad de O(n log n) en el peor de los casos. (LUDA UAM-Azc., s. f.-b)



Ejemplo de HeapSort

Referencias

GeeksforGeeks. (2022, 21 diciembre). Complexity of different operations in binary tree binary search tree and AVL tree. https://www.geeksforgeeks.org/complexity-different-operations-binary-tree-binary-search-tree-avl-tree/

LUDA UAM-Azc. (s. f.-b). http://aniei.org.mx/paginas/uam/CursoAA/curso_aa_20.html