

Instituto Tecnológico y de Estudios Superiores de Monterrey
TC1031
Programación de Estructuras de Datos y Algoritmos Computacionales
Actividad Integradora 2
Conceptos Básicos y Algoritmos Fundamentales
Ing. David Cantú Delgado
Alumna:
Andrea Catalina Fernández Mena
A01197705

Por medio de la siguiente actividad me fue posible comprender elementos claves de estructura y funcionamiento para desarrollar estructuras de datos lineales eficientes. Dichas estructuras de datos nos generan de una manera más automatizada mayor control con cada una de las etapas del proceso de desarrollo de nuestro código como fue el caso del proyecto integrador, en dónde personalmente me fue mucho más sencillo conocer el proceso de organización por medio de etapas de desarrollo de código de la siguiente manera:

1. El desarrollo de “bitácora.h” para obtener el key de conversión de cada registro y así obtener un posible método tangible de jerarquización.
2. Después se genera la estructura del nodo que será empleado para el desarrollo de dicho esqueleto que será utilizado para estructurar los valores posteriores basándose en su jerarquía
3. Una vez estructurado el Nodo se pasa al archivo de “DoubleLinedList.h” en el cuál se emplea dicho esqueleto de nodo ya desarrollado y se colocan a manera de nodo los IP address aún sin ordenar para así lograr que tengan un espacio dentro del Nodo y un espacio de memoria asignado.
4. Posteriormente teniendo dicho nodo, con los métodos de ordenamiento quick sort y binary search dichos valores los ordena
5. Y por último, se espera que por medio de un método stack o queue los almacena.

El desarrollo de ésta lógica por mi cuenta me fue de suma utilidad para el desarrollo y elaboración de mi código. Y creo posible que dicha administración y funciones de búsqueda de datos, pueden ser de suma utilidad para así lograr ordenar o manipular para la búsqueda de información dentro de una base de datos específica.

Así también, me gustó mucho el nivel de dificultad al implicar que utilizáramos Double Linked List para la elaboración del método ya que tiene muchas más ventajas y facilidades para su desarrollo arquitectónico a futuro como se muestra en la siguiente tabla a continuación comparado con la LinkedList Sencilla

Sr. No.	Key	Singly linked list	Doubly linked list
1	Complexity	In singly linked list the complexity of insertion and deletion at a known position is $O(n)$	In case of doubly linked list the complexity of insertion and deletion at a known position is $O(1)$
2	Internal implementation	In singly linked list implementation is such as where the node contains some data and a pointer to the next node in the list	While doubly linked list has some more complex implementation where the node contains some data and a pointer to the next as well as the previous node in the list
3	Order of elements	Singly linked list allows traversal elements only in one way.	Doubly linked list allows element two way traversal.
4	Usage	Singly linked list are generally used for implementation of stacks	On other hand doubly linked list can be used to implement stacks as well as heaps and binary trees.
5	Index performance	Singly linked list is preferred when we need to save memory and searching is not required as pointer of single index is stored.	If we need better performance while searching and memory is not a limitation in this case doubly linked list is more preferred.
6	Memory consumption	As singly linked list store pointer of only one node so consumes lesser memory.	On other hand Doubly linked list uses more memory per node(two pointers).

Por último me gustaría decir que gracias a esta clase he desarrollado y ampliado por muchos mis skills de programación orientada a objetos así como el desarrollo de lógica computacional para entrevista de trabajo, por lo que me siento emocionada de seguir aprendiendo al respecto de nuevas estructuras, y elementos que pueden servir para mi desarrollo profesional como ingeniera en tecnologías computacionales.