Tecnológico de Monterrey

Actividad 3.4 Actividad Integral de BST

06.11.2021

—

Programación de estructuras de datos y algoritmos fundamentales
Grupo 12

Alumno

Fernando Cerriteño Magaña - A01702790

Profesor

Dr. Eduardo Arturo Rodríguez Tello

Reflexión

Para esta actividad se opto por generar un código nuevo y simplemente reutilizando algunas partes de los códigos de entregas pasadas, esto es debido a que se trabajó con las IP’s, dejando sin importancia el cómo se almacenan los datos de hora y fecha. La actividad consistió en obtener dos archivos de texto en base a una bitácora dada, todo esto mientras se implementa el tipo de árbol binario Max Heap, el cual almacena datos por tamaño, dejando siempre en raíz al dato más grande.

Lo primero que se realizo para resolver la actividad fue hacer la lectura de la bitácora y poner los datos recuperados en una variable tipo Max Heap, para realizar se le tuvieron que realizar unas modificaciones al código del Max Heap, esto con el propósito de poder almacenar los datos recuperados evitando problemas de compatibilidad, y especificando que los datos a comparar cuando se haga un pop o un push, sean las IP’s, así mismo, se generó un nuevo método en los datos de tipo entrada, para poder obtener la IP en número decimal.

Una vez que se obtuvo el Max Heap se procedió a generar una bitácora la cual tenga los datos ordenados de mayor a menor de la bitácora inicial en base a sus IP’s, este archivo se nombró “bitácora_ordenada.txt”.

Una vez obtenida la bitácora ordenada se procedió a generar otro Max Heap el cual guardaría las entradas en base a cuál es la entrada que más se repite, para hacer esto se genera una variable temporal en la cual se almacenan los datos previos y realiza una comparación para saber cuántas veces se repite.

Finalmente se obtiene un Max Heap el cual tiene ordenadas de mayor a menor las IP’s en base al número de accesos. Por ultimo se genera otro archivo de texto llamado "ips_con_mayor_acceso.txt" en donde se encuentran las IP’s con mayor número de accesos.

En mi opinión la implementación de arboles binarios facilita exponencialmente la forma en la cual se trabaja con los datos, esto es debido a que con el árbol binario tipo Max Heap se puede realizar un ordenamiento de datos de forma sencilla y sin necesidad de generar un algoritmo de ordenamiento, y mientras que puede que no

tenga la mejor complejidad temporal, facilita el como se tiene que tratar a los datos y evita posibles fallas de compatibilidad cuando se tiene que crear un algoritmo de ordenamiento.

En el contexto de la actividad este código nos puede ayudar a observar cuales son las IP’s que pueden resultar más vulnerables, o que necesiten de una revisión para verificar la integridad de los datos, en base a cuales son las IP’s con más accesos.

Bibliografía

Bolívar, A. A. (2017, 11 agosto). Convertir una IP en un número decimal entero. Neo IT Blog. Recuperado 6 de noviembre de 2021, de https://neoitblog.wordpress.com/2017/08/11/convertir-una-ip-en-un-numero-decimal-entero/

GeeksforGeeks. (2021, 15 septiembre). HeapSort. Recuperado 6 de noviembre de 2021, de https://www.geeksforgeeks.org/heap-sort/