

Reflexion BST

Los BST son estructuras de datos que nos pueden ser muy utiles al momento de realizar ciertos tipos de operaciones, debido a que la velocidad de busqueda de ellos es extremadamente rapido, siendo en promedio $O(\log n)$ dependiendo de si el arbol esta balanceado o no.

En este caso, el arbol de busqueda binario cumple una funcion de manera muy eficiente, que es el guardar los datos y ordenarlos al mismo tiempo, que nos da muchos beneficios, ya que esta operacion tarda en promedio $O(n \log n)$ que es una complejidad de tiempo muy competitiva.

Otro beneficio que esta estructura de datos nos brinda es que hay muchas maneras de recorrer el arbol que nos permiten analizar la informacion de diferentes maneras. En este caso especifico, queremos sacar los valores mas grandes que se encuentran en ella, y para ello existe el recorrido Inorder reverso, que pasa por los valores en orden de mayor a menor, y en este caso nos da una complejidad de $O(\log n)$ para conseguir los 5 valores mas grandes y desplegarlos.

Pero esta estructura de datos tiene un problema que puede llegar a ser bastante grande si no utilizamos alguna variante que mitigue este problema, que es el tener un arbol desbalanceado. El tener un arbol desbalanceado es muy perjudicial en todos los aspectos, ya que en caso de que el arbol sea completamente desbalanceado se incrementa la complejidad de tiempo de busqueda a $O(n)$, lo que convierte al BST en una lista ligada, que en cambio significa que perdemos todos los beneficios que conseguimos al utilizar un BST.

Para evitar esto se suelen utilizar variaciones del BST, como el red-black tree y el AVL tree que se mantienen balanceados de manera automatica.

Tambien existen variaciones extra del BST con funcionalidad diferente, por ejemplo, el Splay tree, que despues de una busqueda o insercion hacen al nodo afectado la raiz del arbol, haciendo que sea el equivalente a un stack en un BST, ya que nos permite accesar el dato recien buscado o insertado.

Para descubrir una red infectada hay varias maneras de darse cuenta. Entre estas maneras la mas importante es la irregularidad en los datos, o es decir, que algo en un dato sea muy diferente de los demas. Por ejemplo, digamos que los datos en general se suelen accesar muy pocas veces, digamos que el promedio de intentos de acceso de un grupo de datos es de 2 veces por sesion, pero de la nada nos percatamos de que un dato especifico tiene 900 intentos de acceso en una sesion, y ahi nos damos cuenta de que hay alguien o algo extra tratando de ingresar ese dato y lo bloqueamos o le informamos al dueño de este para proteger la informacion.