

Recopilación de información

- Predecesor:
El sucesor en un árbol binario o AVL es el siguiente mas pequeño que él, por lo que sería hacer una búsqueda primero en el nodo izquierdo, y allí seguir buscando en los nodos derechos hasta que no exista un nodo derecho.
- Sucesor:
El sucesor en un árbol binario o AVL es el siguiente más grande que él, por lo que sería hacer una búsqueda primero en el nodo derecho, y allí seguir buscando en los nodos izquierdos hasta que no exista un nodo izquierdo.
- Árbol binario:
La estructura de árbol binario se caracteriza por que al momento de nosotros querer insertar un nuevo elemento este empezara a comparar si es mayor o menor, por lo que con esta decisión será el hijo izquierdo (menor), o hijo derecho(mayor).
- Árbol AVL:
El árbol AVL comparte algunas cosas con el árbol binario, pero a diferencia de este, el árbol binario en el peor de los casos la búsqueda puede ser $O(n)$. En el árbol AVL se garantiza que la búsqueda sea de $O(\log n)$ ya que la diferencia de altura entre el nodo izquierdo y derecho de todos los nodos, será del valor absoluto de 1.
- Búsqueda binaria:
La búsqueda binaria se utiliza para reducir la complejidad temporal al momento de realizar alguna búsqueda en bases de datos muy grandes, haciendo que la complejidad sea de $O(\log n)$ reduciendo mucho su costo al compararla con una búsqueda lineal

Soluciones creativas

- Alternativa 1:
Hacer un subString del tamaño que llevan escrito y al momento de encontrar una coincidencia empezar a tomar todos los hijos izquierdos del nodo, y mostrarlos en la lista. Esto pararía si la lista se llena (100 personas) o si se dejan de encontrar
- Alternativa 2:
Hacer un subString del tamaño que llevan escrito y al momento de encontrar una coincidencia empezar a tomar todos los hijos derechos del nodo, y mostrarlos en la lista. Esto pararía si la lista se llena (100 personas) o si se dejan de encontrar

- Alternativa 3:
Hacer un subString del tamaño que llevan escrito y al momento de encontrar una coincidencia se hace un preorden tomando el nodo de la coincidencia como la raíz para hacer el preorden, y colocar en la lista las coincidencias hasta un total de 100, o hasta que no hallan más coincidencias.

Transición de las Ideas a los Diseños Preliminares

- Se descarta la alternativa 1 y 2, ya que al momento de la primera coincidencia el tamaño del árbol para 1000,000,000 de personas, el tamaño del árbol sería menos de 29, haciendo que el resultado de la lista solo pudiera llegar hasta esta cantidad de personas, también se descartó, ya que podría haber personas que no cumplan con la condición de ser iguales los primeros subString (Depende de que búsqueda este haciendo pueden llegar hacer números). Otra razón por la que se descartó es que se estaría saltando muchas veces una cantidad de datos muy grandes.

Por lo que la alternativa con la que nos quedamos es la tercera, al hacer un preorden, se buscara en todos los nodos hijos del nodo ya encontrado, por lo que se encontrarán todas las posibilidades, y al limitar nuestra búsqueda solo a 100 personas, la complejidad temporal no quedaría muy alta