

Pregunta 1

a) la estructura de datos podría ser reemplazada por una tabla de hash, en donde con una función de hash compacta, rápida y que distribuya correctamente (la más distribuidas posibles) podamos hacer las operaciones en $O(1)$, las colisiones se deberán hacer con encadenamiento para que sea posible eliminar. al ser aproximadamente 25.000 alumnos es muy caro buscar con ABB, mientras que con una tabla de hash bien distribuida solo debemos buscar en "la lista" el valor entregado por la función para encontrar al alumno que buscamos y luego revisar la lista ligada de encadenamiento

b) inserción.

inserción(A, k, i)

$x \rightarrow h(k)$

$A[x] = \text{celda} \rightarrow \text{lista de valores}$

$\text{celda.append}(k)$

$\text{celda}[k] = i$

busqueda(A, k)

información = \emptyset

$x \leftarrow h(k)$

$A[x] = \text{celda} \rightarrow \text{lista de valores con mismo valor de hash}$

for j in celda

if $j == k$

información = $\text{celda}[k]$

return información

return información

eliminación(A, k)

$x \rightarrow h(k)$

$A[x] = \text{celda} \rightarrow \text{lista de valores con el mismo valor de hash}$

for i in celda

if $i == k$

eliminamos i de la lista

"diccionario"

función de hash = $h(k)$

$i \rightarrow$ información del estudiante

busca el índice en

donde se debería insertar el valor

y lo agrego a la lista de ese índice

por encadenamiento

típicamente

$h(k) = k \bmod m$

con m primo

k es la llave.

busca si el valor k está en el arreglo A , buscando en la lista que está en el índice con la función de hash

* para poder guardar la información de los estudiantes en A , A tendrá listas en sus índices, de todas las encadenamientos y cada uno dentro de esta lista, será tipo diccionario, como el nodo utilizado en ABB

c). Pasar del ABB a tabla de hash, se utilizará la función de inserción hecha anteriormente.

$T \rightarrow$ árbol

$A \rightarrow$ arreglo de hash.

nosite pasar por todos los nodos de búsqueda.

Mover (T, A)

padre \leftarrow T (raíz del árbol)

if padre.left (si tiene hijo izquierdo)

Mover (padre.left, A)

if padre.right (si tiene hijo derecho)

Mover (padre.right, A)

voy a mis hijos hasta que llega

insercion (A, padre.left, padre.right).

