

0.) a.) Vicente Antonio Espinosa González

b.) Ni, me comprometo

Vicente

1.)

a.) Para los libros ligados se podría tener a cada persona, en ningún orden en específico, concatenada con el siguiente de la forma:

persona 1. next = persona 2

Todas las personas tienen un valor next, menos el último, no hay prev.

Luego, cada persona a su vez tiene una lista ligada con los libros que lee, siguiendo los mismos reglas de la lista de personas.

Es posible hacer tanto por libros como por persona, es mucho más eficiente por persona, pero podría ser al revés

ej:

persona x: libro y

libro y. next = libro z

persona { id: int
next: persona x
libros: libro
}

persona 15 . next = persona 16

libro { next: libro

persona 14 . libro = libro 20

libro 20 . next = libro 42

Se usan 30.000 de memoria, pues la lista de usuarios contiene a todos (5000) y cada usuario contiene sus libros, 6 en promedio, por lo tanto $6 \times 5000 = 30000$

b-) El rut se podría agregar a una tabla de hash, la cual tendría en el lugar n el rut de la persona con id n .
(una 5000 de memoria)

persona {
 next: persona
 libro: libro

hash [... , rut , ...]
 \downarrow
 rut de persona n

libro { next: libro

hacer persona de int X :

$$\text{hash} \cdot \text{posición}(X) = M$$

$$\text{persona} \cdot \text{índ}(M) = \text{persona de int } X.$$

c.)

Para el identificador de libro se puede hacer lo mismo que en b, se usa una tabla de tamaño 600 (una 600 de memoria), y se guarda el código de cada libro en el lugar del 'id' del libro

$$\text{Códigos} \cdot \text{hash} = \left(\dots, \underset{\substack{\uparrow \\ \text{código de libro} \\ \text{con id } K}}{\text{código}}, \dots \right)$$

buscando libro con código W

código-hash. posición (W) = a

find libro con id a = libro con
código W .