Readme about the LFU

Data structure: Use three hash table

Unordered_map m1: <key , pair <value,freq>>

Unordered_map m2: <freq , a list of keys(first in first out)>

Unordered_map m1: <key , its position in the list>

Why do I choose unordered_map?

Because I don’ t need ordered data or traversal through the map, just single element access.

Why do I choose list?

Because doubly-linked lists allow constant time erase operations anywhere within the sequence, and I don’ t need direct access to single element.

Readme about the LRU

Data structure: Use one hash table and a double-linked list

Unordered_map m1: <key , pair<value, its position in the list>>

Double-linked list used;<key>

The reason to choose hash table and list is just like the previous LFU.

I write a function “used” to move the key to the front and change its postion in the list if it is used.