Day 56

Code-

class LRUCache

{

public:

unordered\_map<int, list<pair<int, int>>::iterator> map;

list<pair<int, int>> li;

int cap;

LRUCache(int capacity)

{

this->cap = capacity;

}

int get(int key)

{

if (map.find(key) == map.end())

return -1;

auto address = map[key];

auto item = \*address;

li.erase(address);

map[key] = li.insert(li.end(), item);

return item.second;

}

void put(int key, int value)

{

if (map.size() == cap && !map.count(key))

{

map.erase(li.front().first);

li.pop\_front();

}

if (map.count(key))

li.erase(map[key]);

map[key] = li.insert(li.end(), make\_pair(key, value));

}

};

Time complexity – O(n\*m)

Space complexity-O(1)



