LRU Cache 1

The task is to design and implement methods of an LRU cache. The methods are 'get' and 'set' which are defined as follows:
get(x) : Gets the value of the key x if the key exists in the cache otherwise returns -1
set(x,y) : inserts the value if the key x is not already present. If the cache reaches its capacity it should invalidate the least recently used item before inserting the new item. The capacity of the cache (the maximum elements it may contain) is given as an input parameter.

Жокер: Не сме говорили за структурата от данни свързан списък. Моля прочетете за нея
тук: https://www.informatika.bg/lectures/list https://bg.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D1%8A%D0%BA_(%D0%B0%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%BA%D1%82%D0%B5%D0%BD_%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%BD%D0%B8) https://learn.fmi.uni-sofia.bg/pluginfile.php/183669/mod_resource/content/2/SDA_2018-2019_lecture_4.pdf Programirane=++Algoritmi-v2015.pdf (страница 142) (download link: https://www.programirane.org/wp-content/uploads/downloads/2015/09/Programirane=%2b%2bAlgoritmi-v2015.pdf)

Класът std::list от стандартната темпейтна библиотека STL за C++ е имплементация на структурата от данни (двойно)свързан списък. В Java (Викторио) класът за свързан списък е LinkedList. Използувайте комбинация от std::list<pair<int, int>> cache_list за Ключа (key) и стойността (value) и std::map<int, std::list<pair<int, int>>::iterator> cache_map за ключа (key) и референция към елемента в списъка std::list<pair<int, int>> съдържащ стойността (като втори елемент .second), съответстваща на дадения ключ. В никакъв случай не индексирайте (оператор []) в списъка cache_list и не използвайте find в него, това е ултра бавно и обезмисля използуването му. Ключа трябва също да го има (да е дублиран) в списъка, за да можем да изтрием даден елемент и от мапа, когато го изтриваме от списъка (тогава когато излиза от кеша). В момента, когато кеша е пълен и в него добавите нов елемент отпред, трябва да изкарате последния от него, а това е cache_list.back(). Същевременно трябва да го махнете и от мапа. Но мапа се индексира чрез key. Затова запазването на key е важно и ще го получите от cache_list.back().first

Като цяло, в състезателното програмиране почти не се срещат задачи, чието решение задължително да изисква използването на свързан списък (а не vector). Тази е една от малкото такива. Макар и познаването на свързан списък да изглежда излишно за състезателното програмиране, то структурата свързан списък е една от базовите структури от данни (дърветата и графовете са просто разклонени варианти на линейния свързан списък) в науката информатика (алгоритмичното програмиране), чието познаване е важно за цялостното и разбиране (на информатиката).

Тестовете на тази задача са прости и биха могли да бъдат минати с какво да е решение. Моля опитайте се да го направите с най-ефективното описано в жокера.

Input Format

Each test case contains 3 lines. The first line of input contains an integer N denoting the capacity of the cache and then in the next line is an integer Q denoting the no of queries. Then Q queries follow. A Query can be of two types
1. SET x y : sets the value of the key x with value y
2. GET x : gets the key of x if present else returns -1.

Constraints

1<=N<=10
1<=Q<=100

Output Format

For each test case in a new line output will be space separated values of the query.

Sample Input 0

```
2
5
SET 1 2 SET 1 3 GET 1 SET 2 1 GET 2
```

Sample Output 0

```
3 1
```