Description | My Submissions | Hints/Editorial | AC Submissions | My Notes (0)

LRU Cache

? Ask Doubt

Time-Limit: 2 sec Score: 0/100 Difficulty : ★★★★

Memory: 256 MB Accepted Submissions: 100 Relevant For: AZ-201

Description

Design a data structure that follows the constraints of a **Least Recently Used (LRU) cache**.

Input Format

Implement the LRUCache class:

- **LRUCache(int capacity)** : Initialize the LRU cache with **positive** size capacity.
- **int get(int key)** : Return the value of the key if the key exists, otherwise return -1.
- **void put(int key, int value)** : Update the value of the key if the key exists. Otherwise, add the key-value pair to the cache. If the number of keys exceeds the capacity from this operation, **evict** the least recently used key.

Output Format

For every call to the *get* function print the value of the key if it exists else print -1.

Constraints

$1 \leq capacity \leq 10^5$

$1 \leq key, value \leq 10^5$

At most 10^5 calls will be made to functions *get* and *put*.

Sample Input 1

Copy

```
9 2
put 1 1
put 2 2
get 1
put 3 3
get 2
put 4 4
get 1
get 3
get 4
```

Sample Output 1

Copy

C++14[GCC]

Submit

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  class LRUCache
```