

Homework 11: LRU and LFU Cache

Yuchen Cheng (rudeigerc@sjtu.edu.cn) is responsible for the homework.

Problem

In this homework, your job is to write two small programs to implement **LRU** and **LFU** cache, including following methods: `get(key)` and `put(key, value)`.

LRU, or **Least Recently Used** algorithm, and **LFU**, or **Least Frequently Used** algorithm are both cache algorithm used to manage memory within a computer. Please refer to Wikipedia to get the exact definition, and you would learn more about cache algorithms in CSAPP in your further study.

LRU

```
1  class LRUCache {
2  public:
3      LRUCache(int capacity) {
4          // TODO: your code here
5      }
6
7      int get(int key) {
8          // TODO: your code here
9      }
10
11     void put(int key, int value) {
12         // TODO: your code here
13     }
14 };
```

LFU

```
1  class LFUCache {
2  public:
3      LFUCache(int capacity) {
4          // TODO: your code here
5      }
6
7      int get(int key) {
8          // TODO: your code here
9      }
10
11     void put(int key, int value) {
```

```
12         // TODO: your code here
13     }
14 };
```

Requirement

Design and implement data structure of LRU and LFU cache, and try doing both operations in $O(1)$ time complexity (optional).

`get(key)` Get the value of the key if the key exists in the cache, otherwise return -1.

`put(key, value)` Set or insert the value if the key is not already present. When the cache reaches its capacity, it should invalidate *the least recently used* or *the least frequently used* item before inserting a new item. When there is a tie in LFU, *the least recently used* key would be evicted.

Grading

- LRU: 40%
 - Test cases.
- LFU: 40%
 - Test cases.
- **Documentation: 20%**
 - Please describe your design of LRU and LFU cache in a document, including the time complexity and space complexity of your design. *You would receive more score if you implement methods with $O(1)$ time complexity.*

Hand-in

The files you should hand-in are `lru.h`, `lfu.h` and `*.pdf` (Your documentation should be exported as a PDF file).

Please execute `./handin.sh -id [Your Student ID]` on **macOS** or **Linux**, or you could execute on **Windows** if you have installed some specific softwares, and then submit to the course website. Use `./handin.sh -h` if you are not sure to the command.

Note

- **Submission without documentation would not be judged.**
- **Cheaters would receive zero point.**

Deadline

2018-11-28 17:59:59