

LRU Cache (/problems/lru-cache/)

Submission Detail

18 / 18 test cases passed.

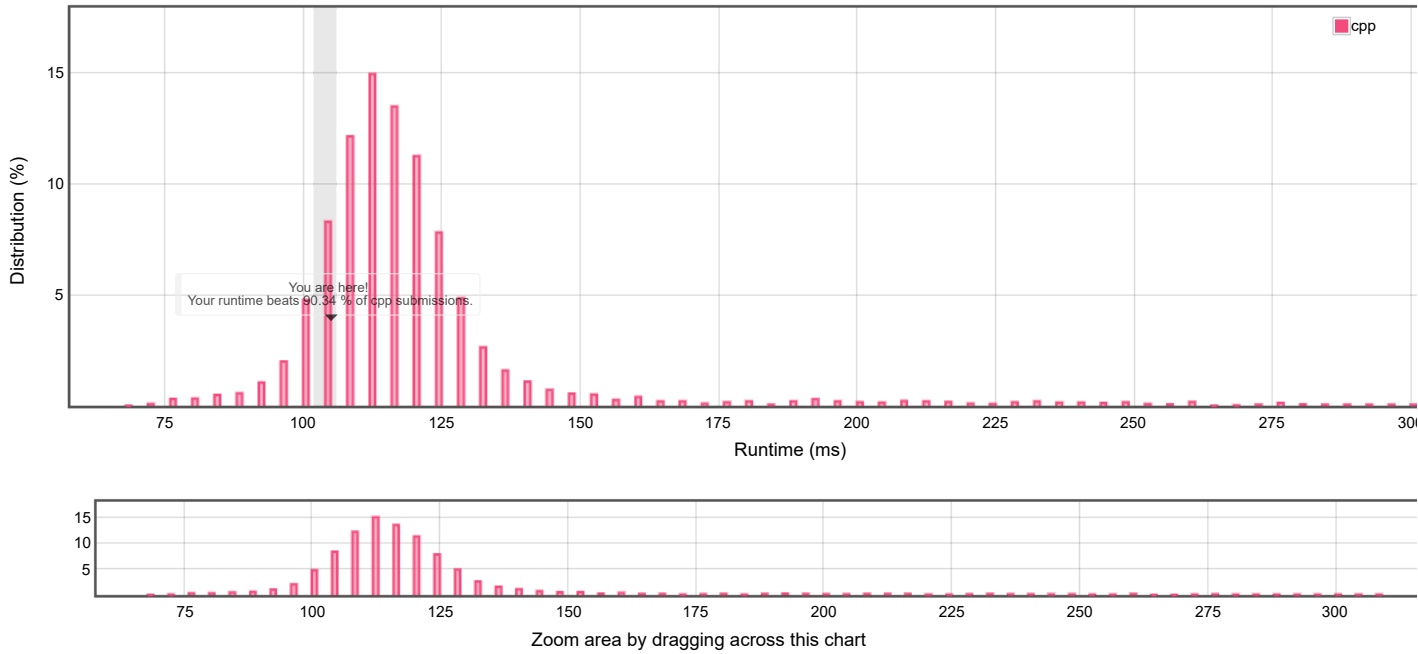
Runtime: 104 ms

Memory Usage: 40.1 MB

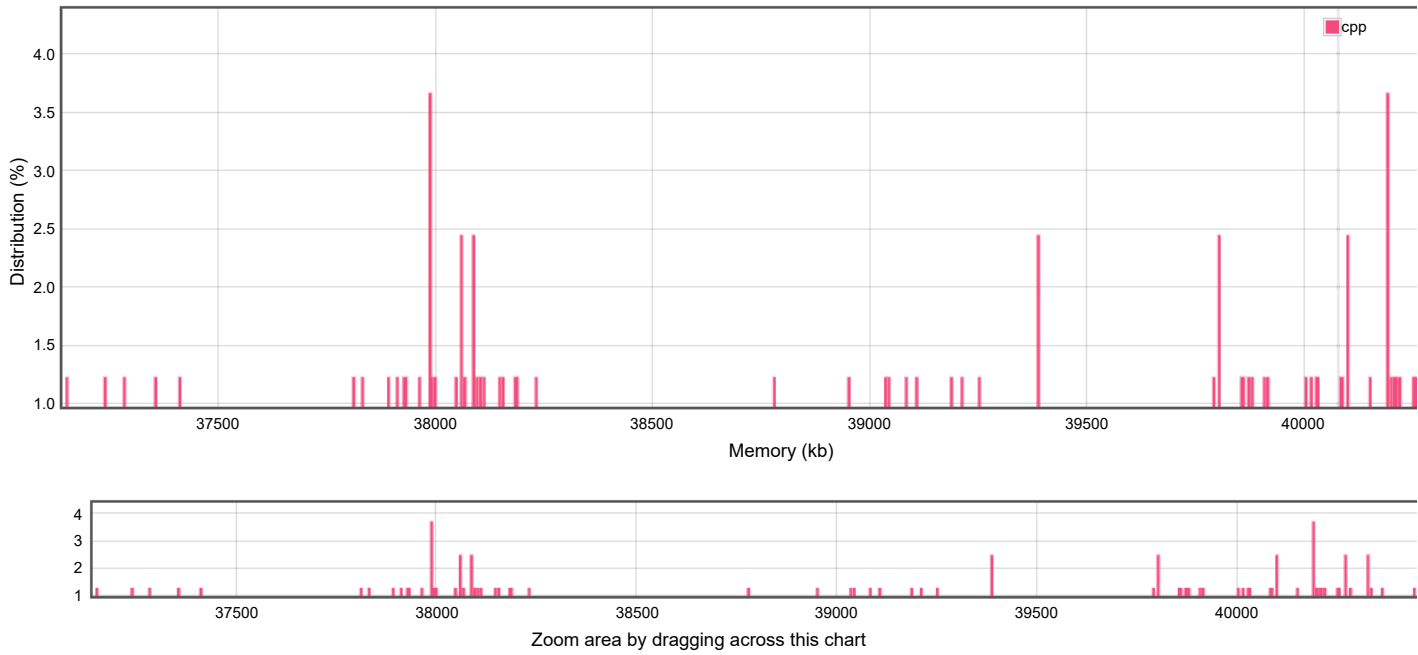
Status: Accepted

Submitted: 1 hour, 11 minutes ago

Accepted Solutions Runtime Distribution



Accepted Solutions Memory Distribution



Invite friends to challenge LRU Cache

Submitted Code: 1 hour, 11 minutes ago

Language: cpp

Edit Code

```
1 #include <unordered_map>
```

```
2 #include <list>
3
4 class LRUCache {
5 public:
6     LRUCache(int capacity) : max{capacity} {
7
8     }
9
10    int get(int key) {
11        auto it = data.find(key);
12        if(it == data.end()) {
13            return -1;
14        } else {
15            return move_to_end(it);
16        }
17    }
18
19    void put(int key, int value) {
20        insert(key, value);
21    }
22 private:
23     struct kv
24     {
25         int key;
26         int value;
27
28         kv(int k, int v) : key{k}, value{v}
29         {
30         }
31     };
32
33     int max;
34     std::list<kv> lru;
35     using iterator = decltype(lru)::iterator;
36     std::unordered_map<int, iterator> data;
37     using map_iterator = decltype(data)::iterator;
38
39
40
41     void remove_if_full()
42     {
43         if(data.size() >= max) {
44             data.erase(lru.front().key);
45             lru.pop_front();
46         }
47     }
48
49     void insert(int key, int value)
50     {
51         auto it = lru.emplace(lru.end(), key, value);
52
53         auto dit = data.find(key);
54         if(dit != data.end()) {
55             auto kvit = dit->second;
56             lru.erase(kvit);
57             dit->second = it;
58         } else {
59             remove_if_full();
60             data.emplace_hint(dit, key, it);
61         }
62     }
63
64     int move_to_end(map_iterator it)
65     {
66         auto newit = lru.emplace(lru.end(), it->second->key, it->second->value);
67         lru.erase(it->second);
68         it->second = newit;
69         return it->second->value;
70     }
71
72 }
```

[Back to problem \(/problems/lru-cache/\)](/problems/lru-cache/)