

LeetCode #2349

Design a Number Container System

The data structure we are going to implement, has two methods:

- change: insert or replace a number at the given index in the system
- find: return the smallest index for the given number in the system

This can be achieved using two hash maps. The first one maps each index to its corresponding value. The second one maps each value to a collection of indices where that value occurs. A sorted data structure such as a sorted map or a heap can be used to maintain the indices in order for the second hash map.

Hash Table key-value to value-key and Sorted Map

indices

key	value

nums

key	value

indices - maps an index to its corresponding value

nums - maps a value to a sorted map of indices

Hash Table key-value to value-key and Sorted Map

indices

key	value

nums

key	value

query - find(10)

there is no value which equals to 10, so the answer is -1

Hash Table key-value to value-key and Sorted Map

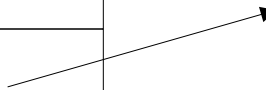
indices

key	value
2	10

nums

key	value
10	

2



query - change(2, 10)

update indices with key 2 and value 10, and add index 2 to the list of indices for value 10 in nums

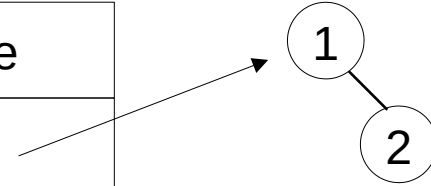
Hash Table key-value to value-key and Sorted Map

indices

key	value
2	10
1	10

nums

key	value
10	



query - change(1, 10)

update indices with key 1 and value 10, and add index 1 to the list of indices for value 10 in nums

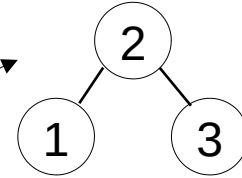
Hash Table key-value to value-key and Sorted Map

indices

key	value
2	10
1	10
3	10

nums

key	value
10	



query - change(3, 10)

update indices with key 3 and value 10, and add index 3 to the list of indices for value 10 in nums

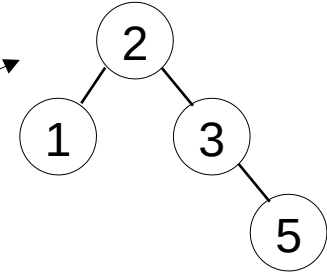
Hash Table key-value to value-key and Sorted Map

indices

key	value
2	10
1	10
3	10
5	10

nums

key	value
10	



query - change(5, 10)

update indices with key 5 and value 10, and add index 5 to the list of indices for value 10 in nums

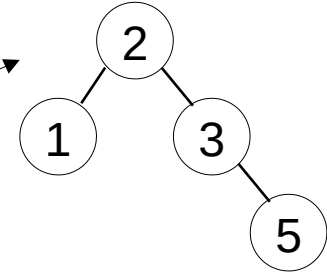
Hash Table key-value to value-key and Sorted Map

indices

key	value
2	10
1	10
3	10
5	10

nums

key	value
10	



query - find(10)

value 10 is present in nums, so we must find the smallest index. That index is 1, so the answer is 1.

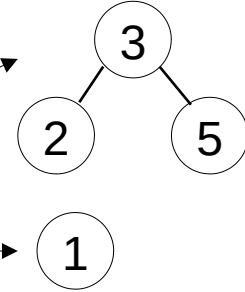
Hash Table key-value to value-key and Sorted Map

indices

key	value
2	10
1	20
3	10
5	10

nums

key	value
10	
20	



query - change(1, 20)

update indices with key 1 and value 20, and add index 1 to the list of indices for value 20 in nums. Since index 1 is present in the list for key 10 and is no longer valid, we have to delete it.

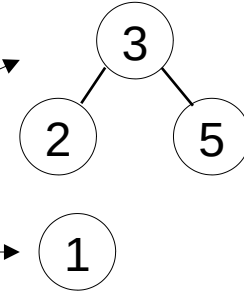
Hash Table key-value to value-key and Sorted Map

indices

key	value
2	10
1	20
3	10
5	10

nums

key	value
10	
20	



query - find(10)

value 10 is present in nums, so we must find the smallest index. That index is 2, so the answer is 2.

Hash Table key-value to value-key and Heap with lazy updates

indices

key	value

nums

key	value

indices - maps an index to its corresponding value

nums - maps a value to a heap of indices

Hash Table key-value to value-key and Heap with lazy updates

indices

key	value

nums

key	value

query - find(10)

there is no value which equals to 10, so the answer is -1

Hash Table key-value to value-key and Heap with lazy updates

indices

key	value
2	10

nums

key	value
10	[2]

query - change(2, 10)

update indices with key 2 and value 10, and add index 2 to the list of indices for value 10 in nums

Hash Table key-value to value-key and Heap with lazy updates

indices

key	value
2	10
1	10

nums

key	value
10	[1, 2]

query - change(1, 10)

update indices with key 1 and value 10, and add index 1 to the list of indices for value 10 in nums

Hash Table key-value to value-key and Heap with lazy updates

indices

key	value
2	10
1	10
3	10

nums

key	value
10	[1, 2, 3]

query - change(3, 10)

update indices with key 3 and value 10, and add index 3 to the list of indices for value 10 in nums

Hash Table key-value to value-key and Heap with lazy updates

indices

key	value
2	10
1	10
3	10
5	10

nums

key	value
10	[1, 2, 3, 5]

query - change(5, 10)

update indices with key 5 and value 10, and add index 5 to the list of indices for value 10 in nums

Hash Table key-value to value-key and Heap with lazy updates

indices

key	value
2	10
1	10
3	10
5	10

nums

key	value
10	[1, 2, 3, 5]

query - find(10)

value 10 is present in nums, so we must find the smallest index. That index is 1, so the answer is 1. During the search for the index we update the heap. While the number is not equal to the value at the key of the first element in the heap (from the indices map), we pop it. Since all elements are up-to-date, this step is unnecessary.

Hash Table key-value to value-key and Heap with lazy updates

indices

key	value
2	10
1	20
3	10
5	10

nums

key	value
10	[1, 2, 3, 5]
20	[1]

query - change(1, 20)

update indices with key 1 and value 20, and add index 1 to the list of indices for value 20 in nums.

Hash Table key-value to value-key and Heap with lazy updates

indices

key	value
2	10
1	20
3	10
5	10

nums

key	value
10	[2, 3, 5]
20	[1]

query - find(10)

value 10 is present in nums, so we must find the smallest index. That index is 1, so the answer is 1. During the search for the index we update the heap. While the number is not equal to the value at the key of the first element in the heap (from the indices map), we pop it. The value associated with key 1 is 20, so we need to pop it.