

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class LRUCache {
4   constructor(maxSize) {
5     this.cache = {};
6     this.maxSize = maxSize || 1;
7     this.currentSize = 0;
8     this.listOfMostRecent = new DoublyLinkedList();
9   }
10
11 // O(1) time | O(1) space
12 insertKeyValuePair(key, value) {
13   if (!(key in this.cache)) {
14     if (this.currentSize === this.maxSize) {
15       this.evictLeastRecent();
16     } else {
17       this.currentSize++;
18     }
19     this.cache[key] = new DoublyLinkedListNode(key, value);
20   } else {
21     this.replaceKey(key, value);
22   }
23   this.updateMostRecent(this.cache[key]);
24 }
25
26 // O(1) time | O(1) space
27 getValueFromKey(key) {
28   if (!(key in this.cache)) return null;
29   this.updateMostRecent(this.cache[key]);
30   return this.cache[key].value;
31 }
32
33 // O(1) time | O(1) space
```

```
1 // Test 1: Insert and Get
2 // Input: maxSize = 2, key = "a", value = 1
3 // Output: 1
4
5 // Test 2: Insert and Get
6 // Input: maxSize = 2, key = "a", value = 1
7 // Output: 1
8
9 // Test 3: Insert and Get
10 // Input: maxSize = 2, key = "a", value = 1
11 // Output: 1
```

Solution 1 Solution 2 Solution 3

```
1 // Do not edit the class below except for the insertKeyValuePair,
2 // getValueFromKey, and getMostRecentKey methods. Feel free
3 // to add new properties and methods to the class.
4 class LRUCache {
5   constructor(maxSize) {
6     this.maxSize = maxSize || 1;
7   }
8
9   insertKeyValuePair(key, value) {
10     // Write your code here.
11   }
12
13   getValueFromKey(key) {
14     // Write your code here.
15   }
16
17   getMostRecentKey() {
18     // Write your code here.
19   }
20 }
21
22 // Do not edit the line below.
23 exports.LRUCache = LRUCache;
24
```

```
1 // Custom Output
2
3
```

