

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.*;
4
5 class Program {
6     static class LRUCache {
7         Map<String, DoublyLinkedListNode> cache = new HashMap<String, Doubl
8         int maxSize;
9         int currentSize = 0;
10        DoublyLinkedList listOfMostRecent = new DoublyLinkedList();
11
12        public LRUCache(int maxSize) {
13            this.maxSize = maxSize > 1 ? maxSize : 1;
14        }
15
16        // O(1) time | O(1) space
17        public void insertKeyValuePair(String key, int value) {
18            if (!cache.containsKey(key)) {
19                if (currentSize == maxSize) {
20                    evictLeastRecent();
21                } else {
22                    currentSize++;
23                }
24                cache.put(key, new DoublyLinkedListNode(key, value));
25            } else {
26                replaceKey(key, value);
27            }
28            updateMostRecent(cache.get(key));
29        }
30
31        // O(1) time | O(1) space
32        public LRUResult getValueFromKey(String key) {
33            if (!cache.containsKey(key)) {
```

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 Program {
5     static class LRUCache {
6         int maxSize;
7
8         public LRUCache(int maxSize) {
9             this.maxSize = maxSize > 1 ? maxSize : 1;
10        }
11
12        public void insertKeyValuePair(String key, int value) {
13            // Write your code here.
14        }
15
16        public LRUResult getValueFromKey(String key) {
17            // Write your code here.
18            return null;
19        }
20
21        public String getMostRecentKey() {
22            // Write your code here.
23            return null;
24        }
25    }
26
27    static class LRUResult {
28        boolean found;
29        int value;
30
31        public LRUResult(boolean found, int value) {
32            this.found = found;
33            this.value = value;
```



1	import sys	10
2	import random	10
3	import time	10
4	import math	10
5	import os	10
6	import sys	10
7	import random	10
8	import time	10
9	import math	10
10	import os	10
11	import sys	10
12	import random	10
13	import time	10
14	import math	10
15	import os	10
16	import sys	10
17	import random	10
18	import time	10
19	import math	10
20	import os	10
21	import sys	10
22	import random	10
23	import time	10
24	import math	10
25	import os	10
26	import sys	10
27	import random	10
28	import time	10
29	import math	10
30	import os	10
31	import sys	10
32	import random	10
33	import time	10
34	import math	10
35	import os	10
36	import sys	10
37	import random	10
38	import time	10
39	import math	10
40	import os	10
41	import sys	10
42	import random	10
43	import time	10
44	import math	10
45	import os	10
46	import sys	10
47	import random	10
48	import time	10
49	import math	10
50	import os	10
51	import sys	10
52	import random	10
53	import time	10
54	import math	10
55	import os	10
56	import sys	10
57	import random	10
58	import time	10
59	import math	10
60	import os	10
61	import sys	10
62	import random	10
63	import time	10
64	import math	10
65	import os	10
66	import sys	10
67	import random	10
68	import time	10
69	import math	10
70	import os	10
71	import sys	10
72	import random	10
73	import time	10
74	import math	10
75	import os	10
76	import sys	10
77	import random	10
78	import time	10
79	import math	10
80	import os	10
81	import sys	10
82	import random	10
83	import time	10
84	import math	10
85	import os	10
86	import sys	10
87	import random	10
88	import time	10
89	import math	10
90	import os	10
91	import sys	10
92	import random	10
93	import time	10
94	import math	10
95	import os	10
96	import sys	10
97	import random	10
98	import time	10
99	import math	10
100	import os	10

Run or submit code when you're ready.