Walkthrough of Code:
To implement a constant time complexity would involve the use of two or more data structures in the cache. One data structure alone can't implement all the methods because of their constraints. A dictionary, given an index, returns the value with a constant time complexity. A dictionary doesn't shift values, once set, they are only able to be popped from their index. A linked list can shift a node without altering the linked list in 0(1) time. It can delete and reset the links of a referenced node and surrounding nodes. The linked list has references to tail and head, replacing a queue. Though a linked list can shift nodes efficiently, it would need a reference to the node without iteration. Thus, either data structure is that of the other's constraints and purposes.

Time Complexity: O(1)
The time complexity is a constant value. It won't vary as the input does. The cache implements all the methods without iterating through the values. Instead, it relies on a dictionary and references to certain nodes to implement a constant time complexity.

Space Complexity: O(n)
The dictionary and the linked list append the inputs given, the index and the value of the nodes entered. The length of the linked list, the dictionary, and the space complexity vary linearly with the input.