

Hash Table

RunSubmit

881

DescriptionEditorSolutionsSubmissions

1756. Design Most Recently Used QueuePremium

Solved

MediumTopicsCompaniesHint

Design a queue-like data structure that moves the most recently used element to the end of the queue.

Implement the MRUQueue class:

MRUQueue(int n) constructs the MRUQueue with n elements: [1,2,3,...,n].

int fetch(int k) moves the kth element (1-indexed) to the end of the queue and returns it.

Example 1:

Input:

["MRUQueue", "fetch", "fetch", "fetch", "fetch"]

[[8], [3], [5], [2], [8]]

Output:

[null, 3, 6, 2, 2]

Explanation:

MRUQueue mRUQueue = new MRUQueue(8); // Initializes the queue to [1,2,3,4,5,6,7,8].

mRUQueue.fetch(3); // Moves the 3rd element (3) to the end of the queue to become [1,2,4,5,6,7,8,3] and returns it.

mRUQueue.fetch(5); // Moves the 5th element (6) to the end of the queue to become [1,2,4,5,7,8,3,6] and returns it.

mRUQueue.fetch(2); // Moves the 2nd element (2) to the end of the queue to become [1,4,5,7,8,3,6,2] and returns it.

mRUQueue.fetch(8); // The 8th element (2) is already at the end of the queue so just return it.

Constraints:

1 <= n <= 2000

1 <= k <= n

At most 2000 calls will be made to fetch.

Follow up:

Finding an O(n) algorithm per fetch is a bit easy. Can you find an algorithm with a better complexity for each fetch call?

Seen this question in a real interview before?

1/5

Yes

No

Accepted

15.2K

Submissions

19.9K

Acceptance Rate

76.0%

Topics

Companies

Hint 1

Hint 2

Hint 3

Similar Questions

Discussion (2)

Copyright © 2024 LeetCode All rights reserved

</> Code

Python3Auto

1class MRUQueue:

2

3def __init__(self, n: int):

4

5self.n = [x for x in range(1, n+1)]

6

7

8def fetch(self, k: int) -> int:

9

10val = self.n[k-1]

11del self.n[k-1]

12self.n.append(val)

13return val

14

15

16

17# Your MRUQueue object will be instantiated and called as such:

18# obj = MRUQueue(n)

19# param_1 = obj.fetch(k)

Saved

Ln 5, Col 44

TestcaseTest Result

AcceptedRuntime: 25 ms

Case 1

Input

["MRUQueue","fetch","fetch","fetch","fetch"]

[[8],[3],[5],[2],[8]]

Output

[null,3,6,2,2]

Expected