Saturday, 10 April 2021 9:02 PM

Q Given an array of integers and a number k, return true / false

if the given array can be divided into pairs such that sum of every pair is divisible by k.

Eg: arr = [9, 7, 5, 3] k = 6

output → true

(9, 3) , (7, 5)
↓ ↓
12 12

Eg: ~~arr~~ → [91, 74, 66, 48] k = 10

output → false

(74, 66) (91, 48)
↓ ✗
140

→ If (arr.length) % 2 != 0 false

given 2 nums ⓐ and ⓑ

a % k == x and b % k == k - x

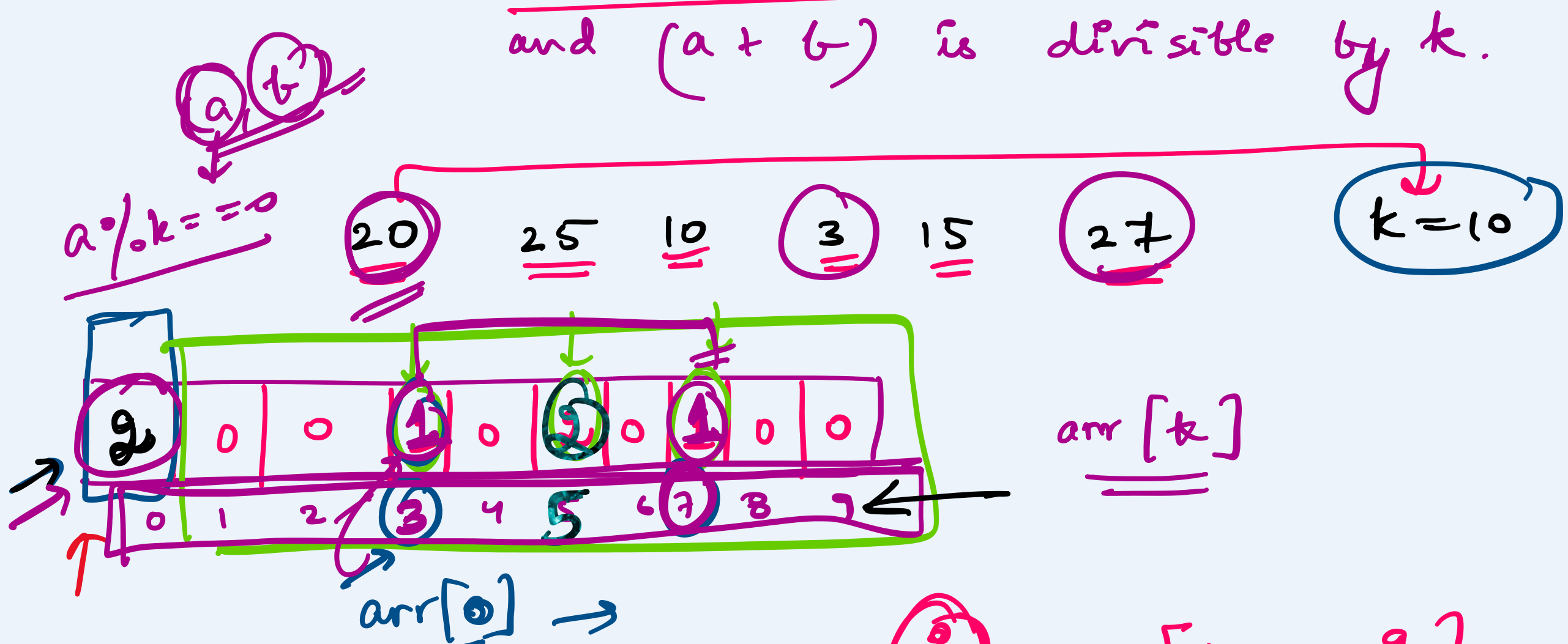
then (a + b) % k == 0

Proof:

a % k == x given
b % k == k - x

(a + b) % k = ((a + b) % k) % k
= (a % k + b % k) % k
= (x + k - x) % k
= k % k = 0

Hence (a + b) % k = 0 → To prove
and (a + b) is divisible by k.



arr[0] →

ⓘ → [1, — 9]

k - i

[20, 10]
[3, 27]
[25, 15]

true

Q Given an arr of size n and integers k, return the distinct no. in all windows of size k.

arr = {1, 2, 1, 3, 4, 2, 3} k = 4

out → | 3 | 4 | 4 | 3 |

[1, 2, 1, 3, 4, 2, 3]

ans = 3, 4, 4, 3

2 → ~~1~~ ~~2~~ 1
3 → ~~1~~ 2
4 → 1

Q Design a DS such that LRU cache

Implement a LRU class here:—

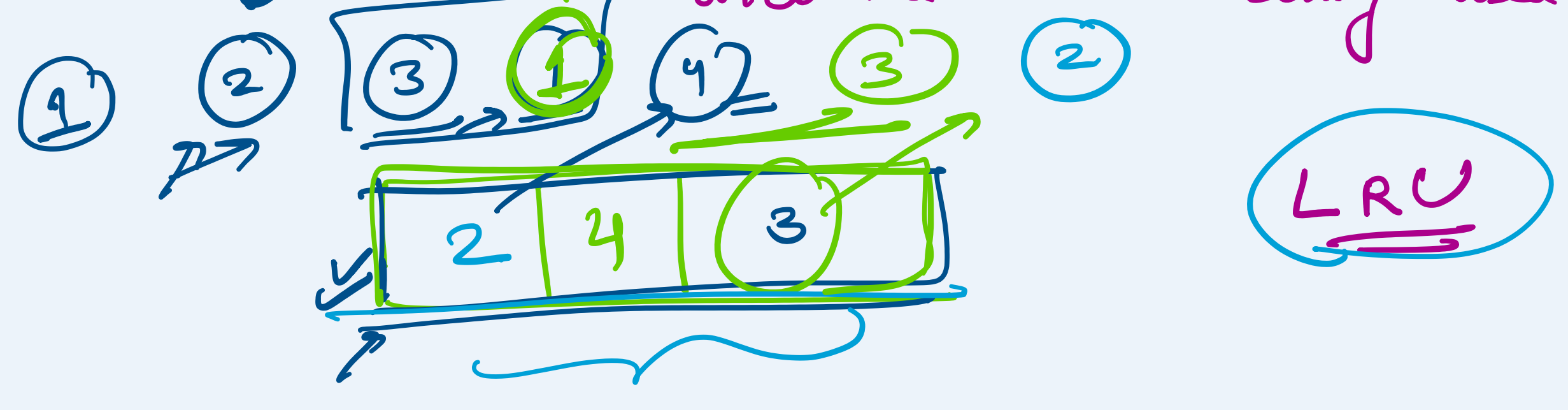
① LRU cache (int capacity) ③

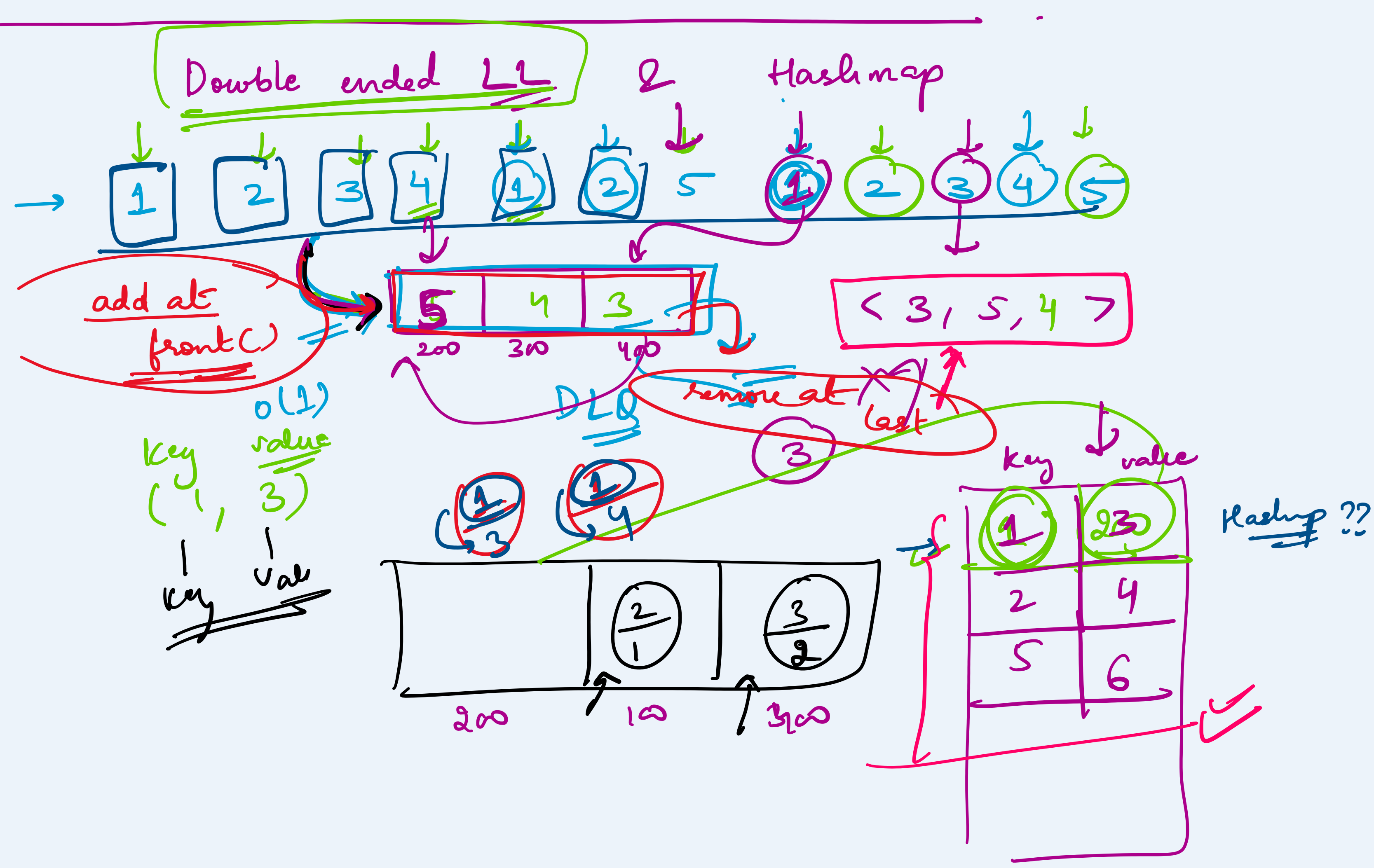
② int get(int key)

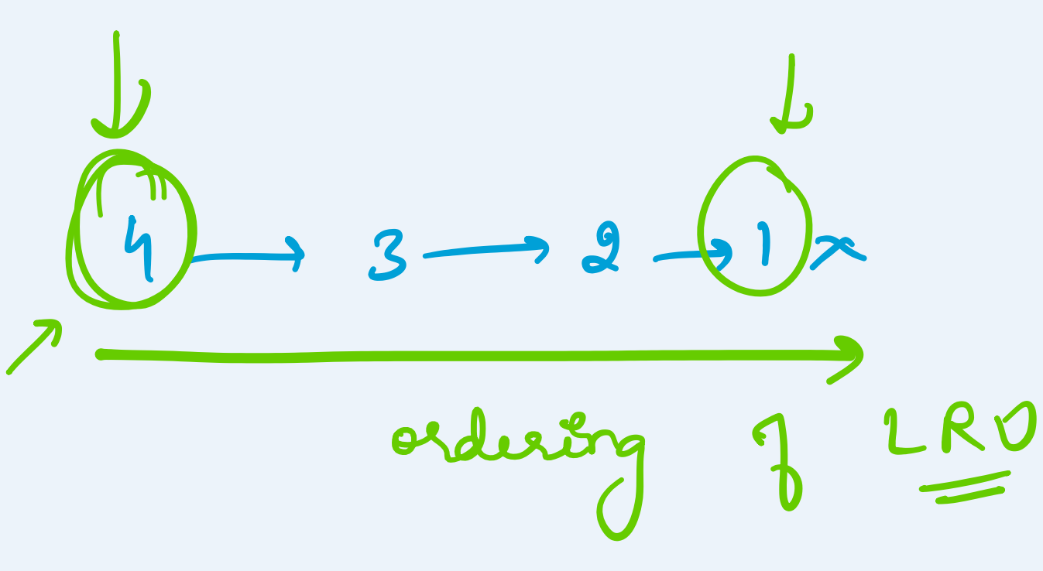
③ void put (key, value)

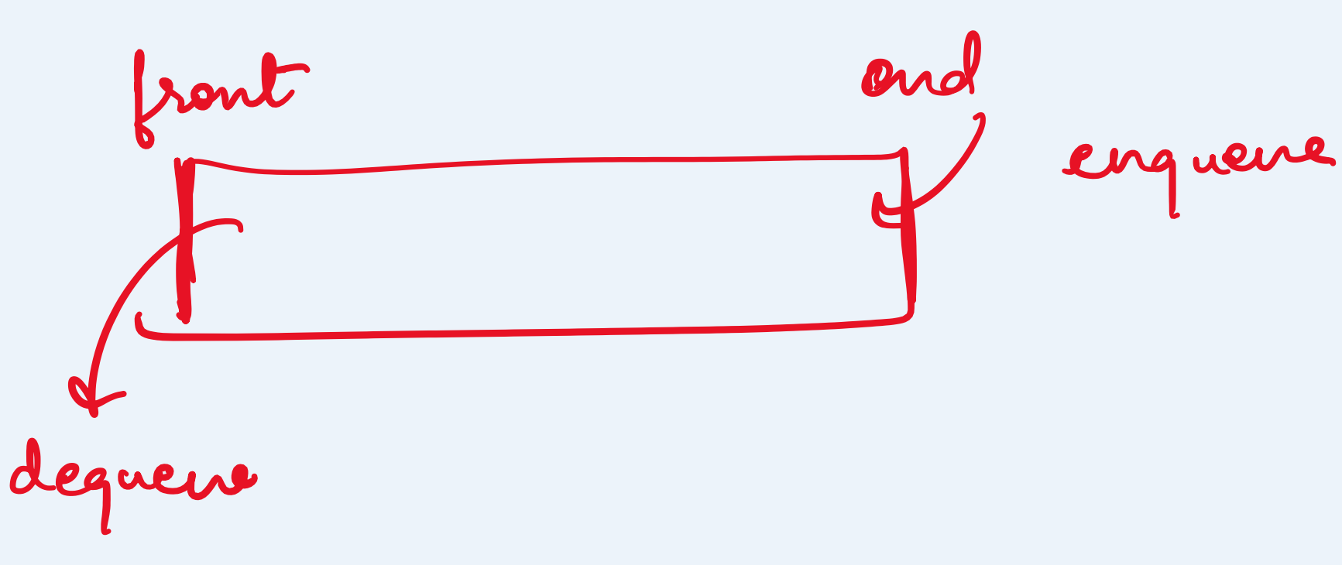
O(1)

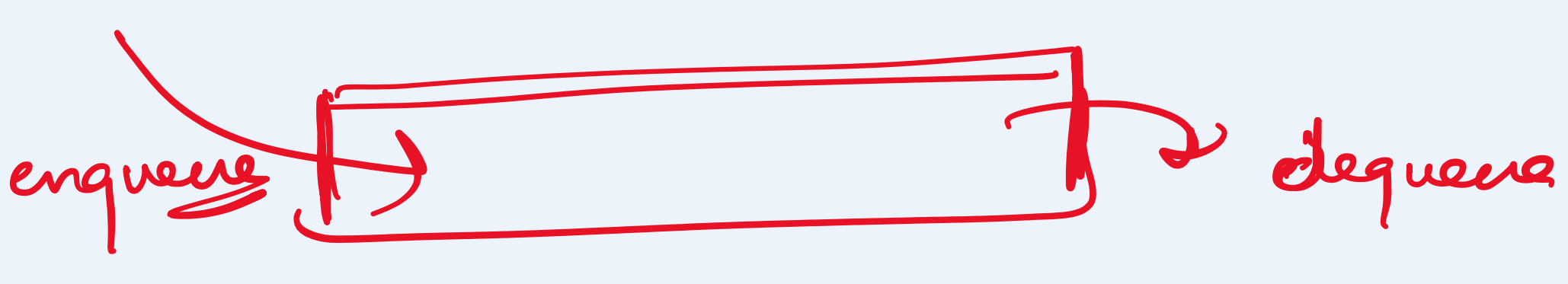
evict the least recently used key











Q Find the length of largest subarray with 0 sum.

arr = [15, -2, 2, -8, 1, 7, 10, 23]

output → 5

arr = [1, 2, 3]

output → 0

prefix(i) = arr[0] + arr[1] + ... arr[i]

prefix[j] = arr[0] + arr[1] + ... arr[j]

If j > i

if (prefix (i) == prefix (j))

then

prefix(j) − prefix(i) = 0

↓

arr[i+1] . . . . a[j] = 0