04 March 2023 09:40

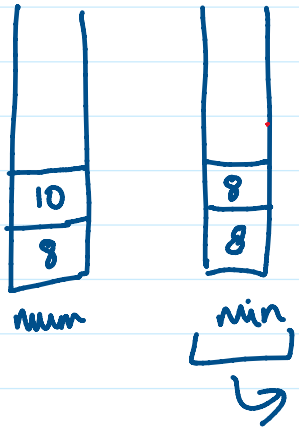
× p min = ∞ 8 5

int min = 8 8 4

1) MinStack
- push() ✓ O(1)
- pop() ✓ O(1)
- peek() ✓ O(1)
- min() → smallest element O(1)

At max m calls

2

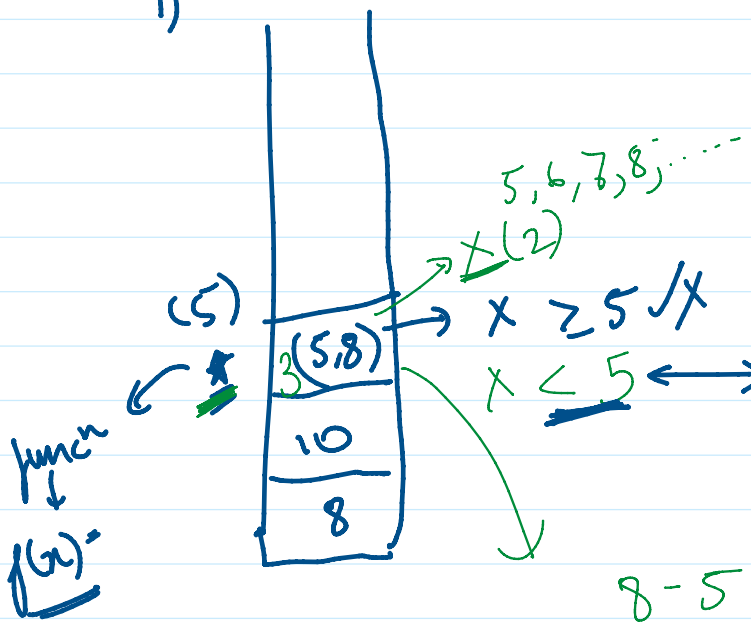


S → O(n)
↳ O(2·n)
✓
×

min peek
8 , 10

1)

min = 8 5



8 - 5 : 3

8
5
2
0
↓
5 → Update min

4 < 5
pop

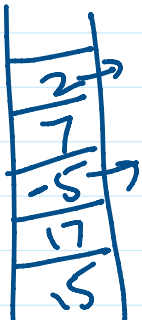
min

5 8

curr < prev

curr - prev < 0

2 curr - prev < curr
(5)
×



min = 8 2

10/2 - 15 = -5

$x = curr + (curr - prev)$

$pop(prev = 2curr - x)$ (min)

$5 = 2 \times 2 - (-1)$

[10, 20, 30, 40, 50]

[50, 40, 30, 20, 10]

10 ([20, 30, 40, 50]) dyso(10) [50, 40, 30, 20, 10]

20, (30, 40, 50)) dyso(20) [50, 40, 30, 20]

(40, 50) dyso(30) [50, 40, 30]

[50, 40] [50]

40, (50) dyso(40) | (.) dyso(50) return

[10, 20, 30, 40, 50]

(10) ([20, 30, 40, 50, 10]) dyso(10)

([30, 40, 50, 10, 20]) dyso(20)

([40, 50, 10, 20, 30]) dyso(30)

([50, 10, 20, 30, 40]) dyso(10)

([10, 20, 30, 40, 50]) dyso(50) BC

50
40
30
20
10

First -ve number in window of size k = 3 -

[20, -5, 10, 9, -8, -2, 3, 6, -4]

-5, -5, -8, -8, -8, -2, -4

for(i=0, n){
  for(j=i ; j<i+k)
}

n * k

↓

O(n+k)

9 - 3 = 6

i → (0, 6)

[20 -5 10 9 -8 -2, 3, 6, -4]
 0 1 2 3 4 5 6 7 8

preprocessing → start extra elements
↳ end window complete (queue size)

5, 8

Remove existing -ve number

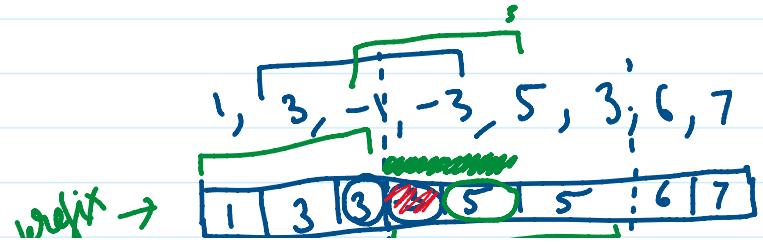
-5, -5, -8, -8, -8, -2, -4

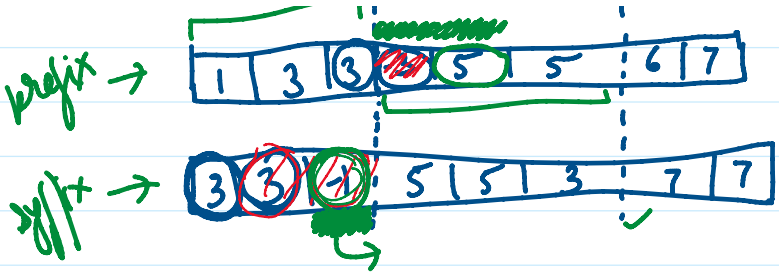
[20 -5 10 9 -8 -2, 3, 6, -4]

[1, 3, -1, -3, 5, 3, 6, 7] k = 3

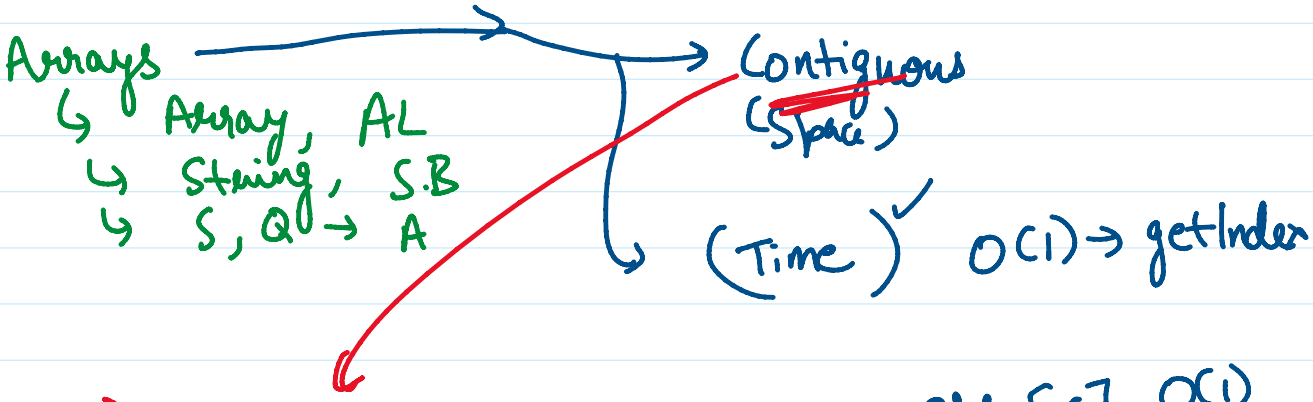
3
3
5
5
6
7

k
↳ Array Dequeue
Array




prefix
1 3 3 5 5 6 7
suffix
3 3 -1 5 5 3 7 7

04 March 2023 12:27



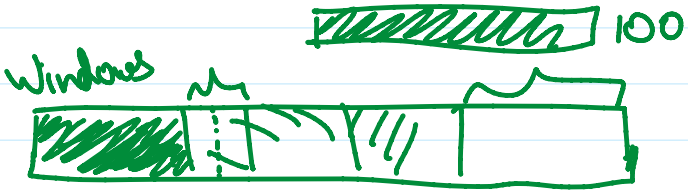
1) Memory Waste

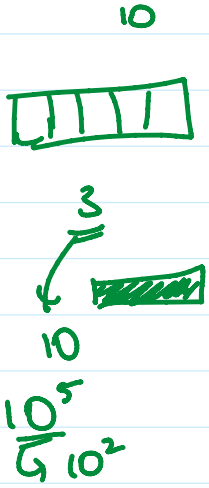
2) Contg. Memory not available

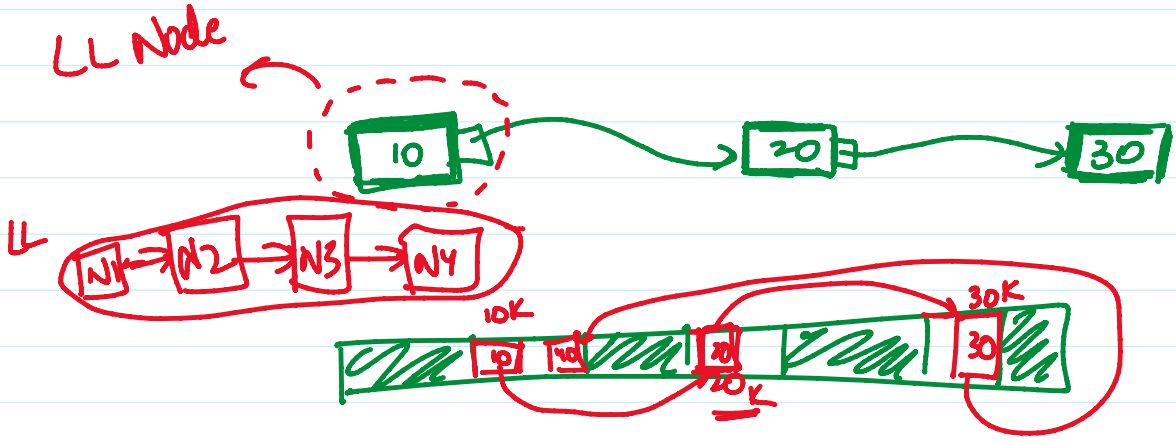
arr[5] O(1)

100

104 X







Contigous??

Linked List

Node {
data;

int *1000

1000 + 4

Node {
data;
Node next;
}

address of next node

1000 + 4