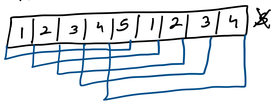
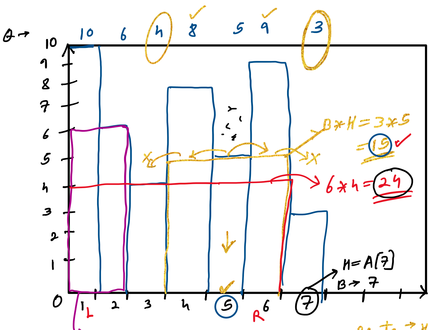
Stacks and Queues

Friday, 24 December 2021 8:59 PM

A → [1|2|3|4|5]

A append A & remove last element





Find the area of largest rectangle that can be formed in given histogram.

Brute → ∀(i,j) find the area of rectangle formed. TC = O(N³) → O(N²)

Area → Base = (R-L+1); Height = min(∀i A[i])

H → A[5] ✓

If A[i] is height, what is the max area rectangle?

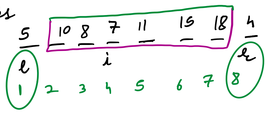
∀i if H = A[i] & max possible Base = B ⇒ Area = (A[i] * B)

O(N) (N) → O(1)

l → index of nearest left min
r → index of nearest right min.
} B = r - l - 1 ✓

r - l - 1 = 8 - 1 - 1 = 6

O(N) using stacks



Ans = max ∀i (A[i] * (r[i] - l[i] - 1))

TC = O(N)
SC = O(N)

```
ans = 0;
for(i = 0 ; i < n ; i++){
    h = A[i];
    for(j = i ; j < n ; j++){
        h = min(h, A[j]);
        ans = max(ans, h * (j - i + 1));
    }
}
return ans;
```

i → j

B = (j - i + 1)

A → [10, 6, 4, 8, 5, 9, 3]

L = min area = 3 * 7 (min element * #elements)

R = max area = 10 * 7 = 70 (max element * #elements)

M = 73/2 = 36

Q → Find first non-repeating character from left in a running stream of characters for every character.

If there is none print # in that position.

Freq Map

a → 2
b → 2
c → 2
d → 1

TC = O(N)
SC = O(N)

S → a b c a c b d

Ans → a a a b b # d

FIFO

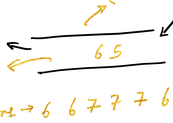
Q → Given an integer array. Find max element ∀ subarrays of size K.

Brute TC = O(N * k)

K = 3

N - K + 1

Insert index in dq instead of elements.



Ans → 6 6 7 7 7 6

↳ Front of queue ✓

1) While rear element < A[i] → pop from rear end. (multiple time)
2) Insert A[i] from rear.
3) If front element is outside the window → dequeueFront();
4) Ans[i] = front element.

TC = O(N) SC = O(K) ✓