Arrays - II

curr = a[i]

Leaders in an Array

a[] = [2, 7, 6, 1, 4, 3]

leaders: the elements which do not have any larger element on their right side.

output : 7, 6, 4, 3

O(N²) → O(N)

loop1 :

```
for (int i = 0; i < n ; i++) {
    cur = a[i]; boolean isLeader = true;
    for (int j = i+1 ; j < n ; j++) {
        if ( a[j] > cur ){
            isLeader = false; break;
        }
    }
    if (isLeader) {
        Print (a[i]);
    }
}
```

$O(N^2)$

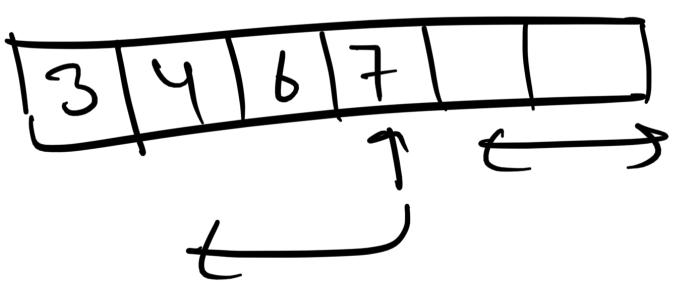
a[] = [2, 7, 6, 4, 1, 3]

largest = 7

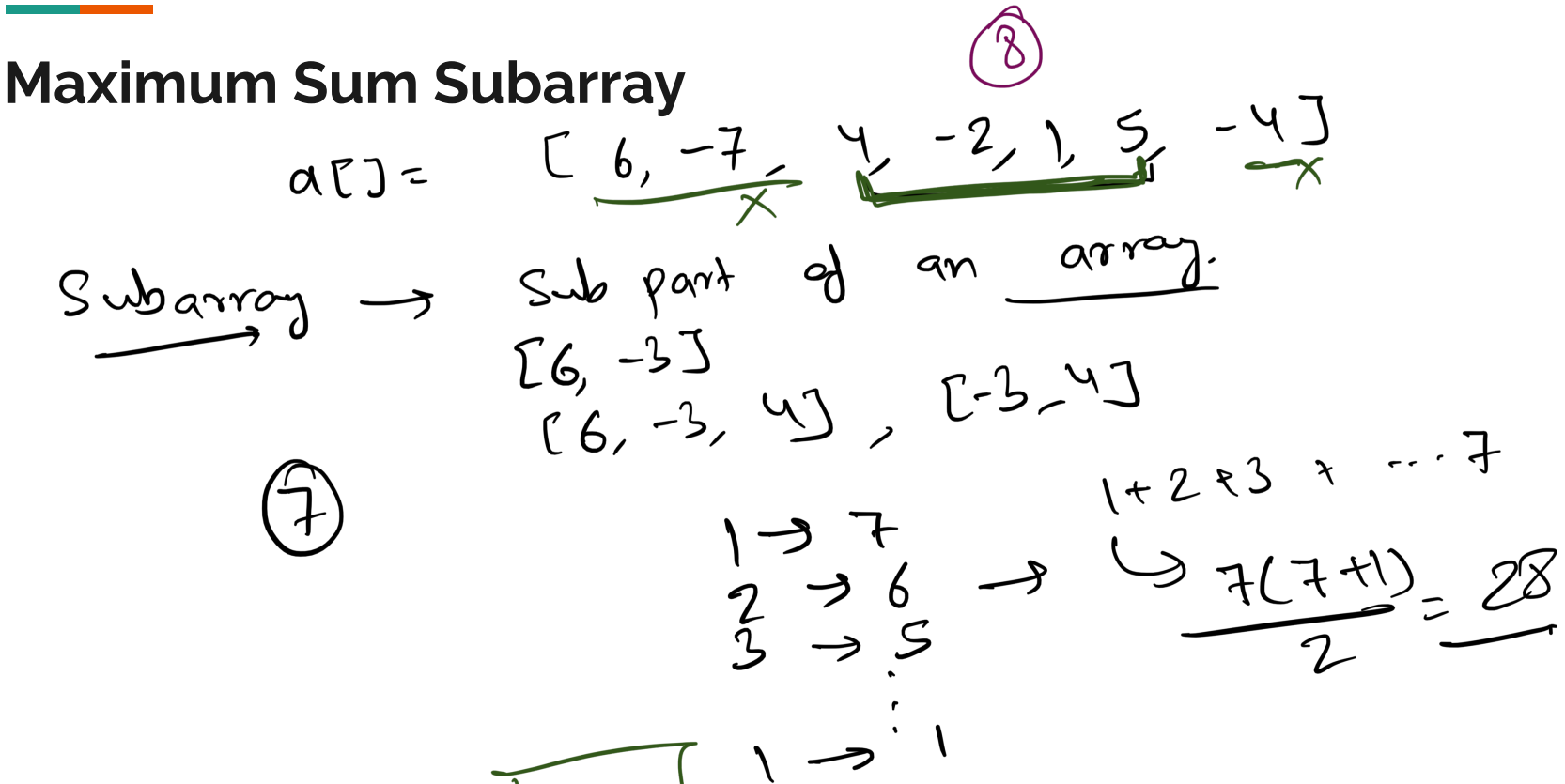
7, 6, 4, 3

3, 4, 6, 7

Create a Array



Maximum Sum Subarray



$\frac{n(n+1)}{2} \rightarrow O(N^2)$

a[] = [6, -7, 4, -2, 1, 5, -4]

int curSum = 0+6 → 6+-7 → -1 → 0+4 → 4+-2 = 2+1

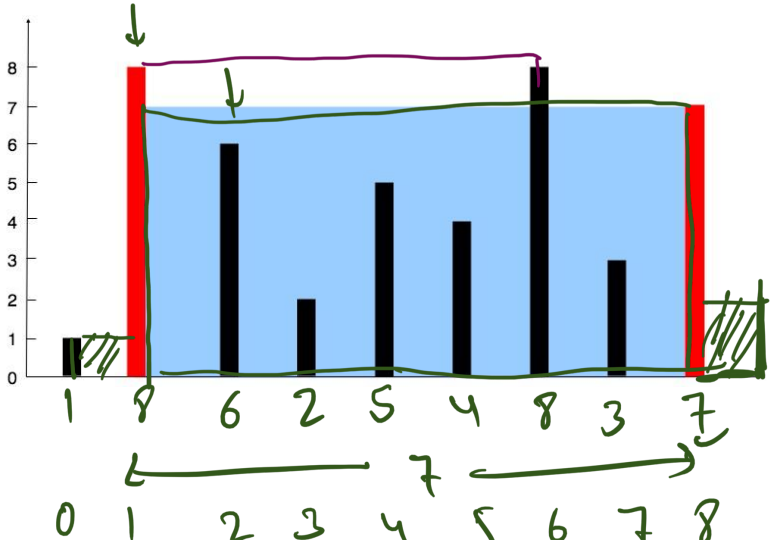
int maxSum = -∞ 6 8

4 ⟷ +8 ⟷ 5+3

```
int cur = 0
int max = -∞
loop () {
    cur += a[i];
    if (cur < 0) cur = 0
    if (cur > max) max = cur
}
```

max

Container with most Water

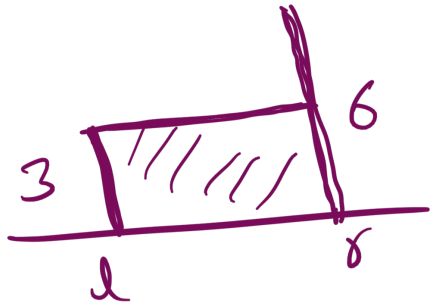


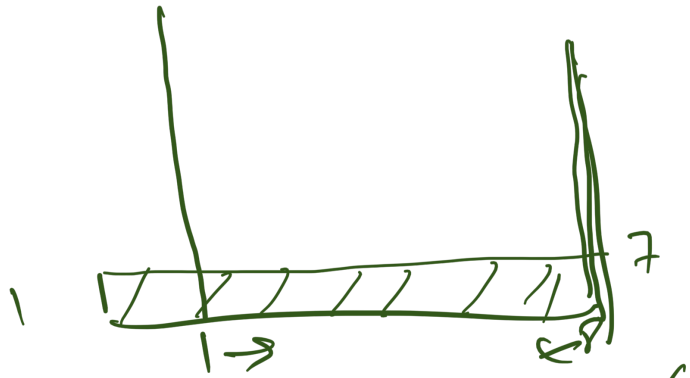
O(N²)

8 × 5 = 40

$O(N^2)$

```
for ( i →  ) {
    int left = a[i];
    for (j = i+1 →) {
        int right = a[j];
        int dist = j - i
        area = min(left, right) * dist
        maxArea = max ( maxArea, area);
    }
}
```





Two pointers approach

a[] = (1, 8, 6, 2, 4, 5, 8, 3, 7)

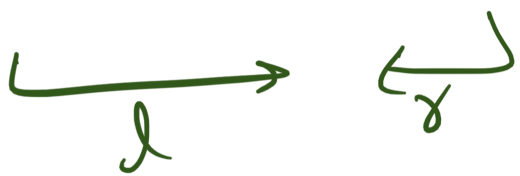
to: 0 1 2 3 4 5 6 7 8

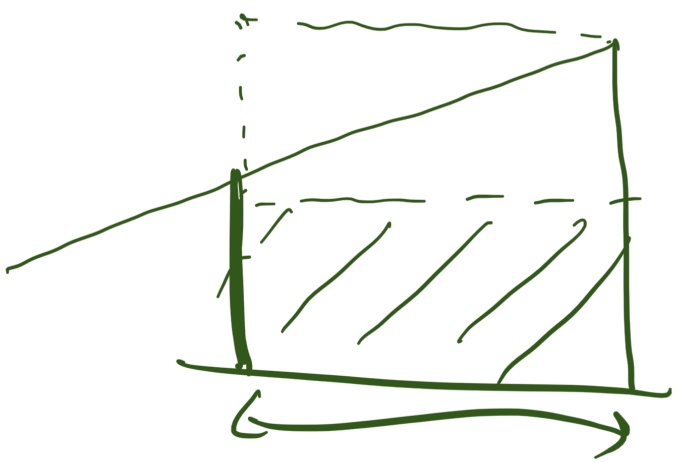
l → 1, r → 8

area = 8 * 1 = 8 → 7 * 7 = 49

maxArea = -∞ → 8 → 49

3 * 6 = 18

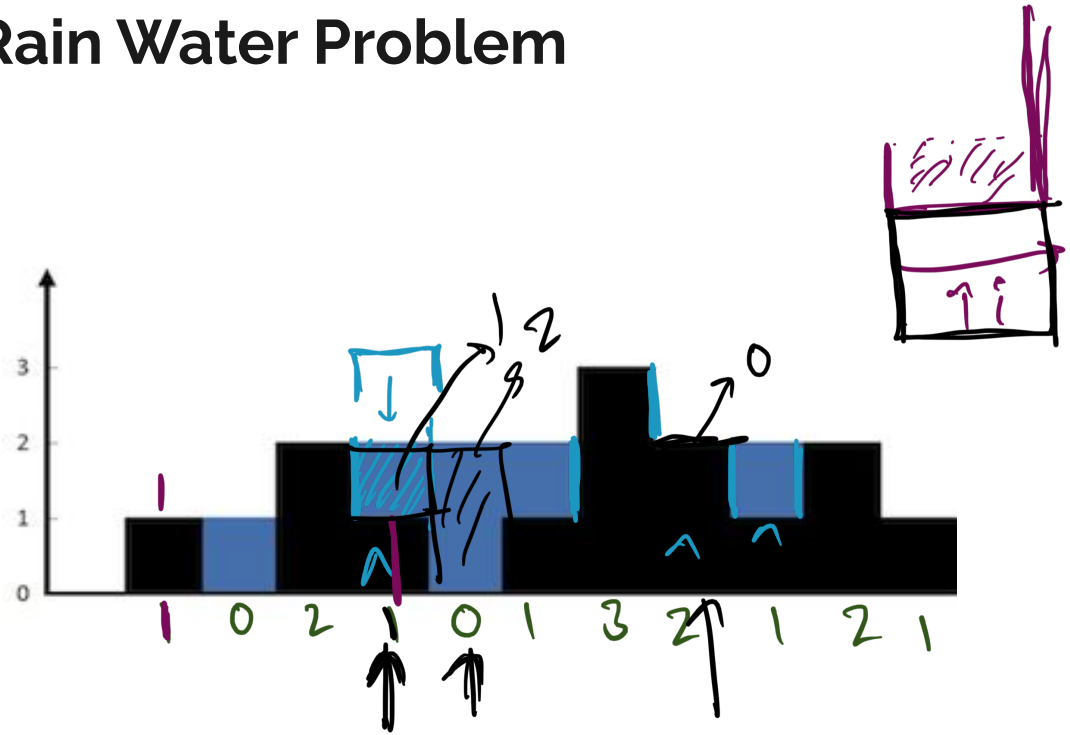




Assignment:

Largest Area in a Histogram

Trapping Rain Water Problem



l[] = 1 1 2 2 2 2 3 3 3 3 3

r[] = 3 3 3 3 3 3 3 2 2 2 1

min (l[i], r[i]) − a[i]

min(2, 3) − 0
2 − 0 = 2

min(2, 3) − 2
2 − 2 = 0

min(2, 3)
2 − 1 = 1

```
int l[0] = a[0];
for (int i = 1; i < n; i++) {
    l[i] = Math.max(a[i], l[i-1]);
}
```

Practice Problems

1. Print frequency of all the elements in a sorted Array.
2. In an Array of all 0s & 1s, find the longest length of all consecutive 1s.

   Input: 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1

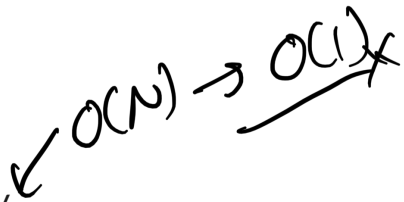
   Output: 4

3. Move all 0s to the end of the given Array.

   Input: 8, 0, 1, 3, 0, 0, 5

   Output: 8, 1, 3, 5, 0, 0, 0

4. Trapping Rain water Problem in O(1) space complexity.
5. Minimum Sum Subarray Problem



Practice Problems

6. Print the elements in the maximum sum subarray .

7. Stocks Buy & Sell problem

    https://leetcode.com/problems/best-time-to-buy-and-sell-stock/

    https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii/

// link of
kadane's subarray