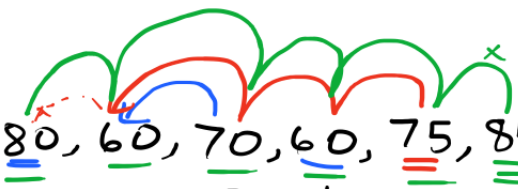


Stock Span Problem

int[] prices = [100, 80, 60, 70, 60, 75, 85]
 0 1 2 3 4 5 6
 { 1 1 1 2 1 4 6 }



Span: no. of consecutive days before current day
where the price \leq price at current day.

Brute Force:

- 1) Using 2 loops count the number of elements before
curr_idx.

TC: $O(N^2)$

SC: $O(1)$



- 2) \rightarrow Using Stack find the index of the

previous element just greater than the current element

{ 100, 80, 60, 70, 60, 75, 85 }

0 ✓ ✓ 1 2 3 4 5 6

{ 1 1 1 2 1 5-1 6-0 }

4 6

TC: O(N)

SC: O(N)



→ empty stack: no elements greater than current

Stack will store the index of greater elements on left side

```
static int[] stockSpan(int[] prices) {
    Stack<Integer> stack = new Stack<>();
    // store idx of greater elements on left

    int n = prices.length;
    int[] res = new int[n];
    for (int i = 0; i < n; i++) {
```

{ 100, 80, 60, 70, 60, 75, 85 }

0 1 2 3 4 5 6



```

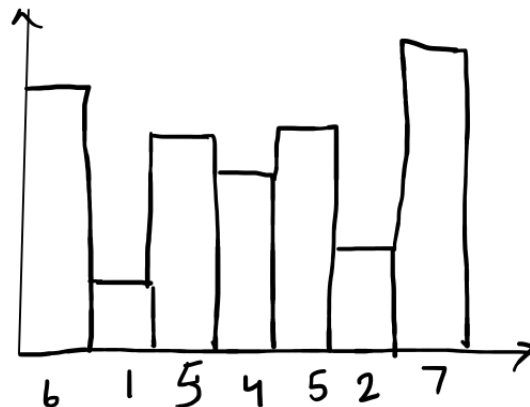
int curr_price = prices[i];
while (!stack.isEmpty() && prices[stack.peek()] <= curr_price) {
    // remove smaller or equal elements from the stack
    stack.pop();
}
if (stack.isEmpty()) {
    // current element is the greatest element so far
    // no elements greater than current element was found on left side
    res[i] = i + 1;
} else {
    // we found a greater element on the left side
    // all the elements between the greater element on left and i
    // are included in the span
    res[i] = i - stack.peek();
}
stack.push(i);
}
return res;
}

```

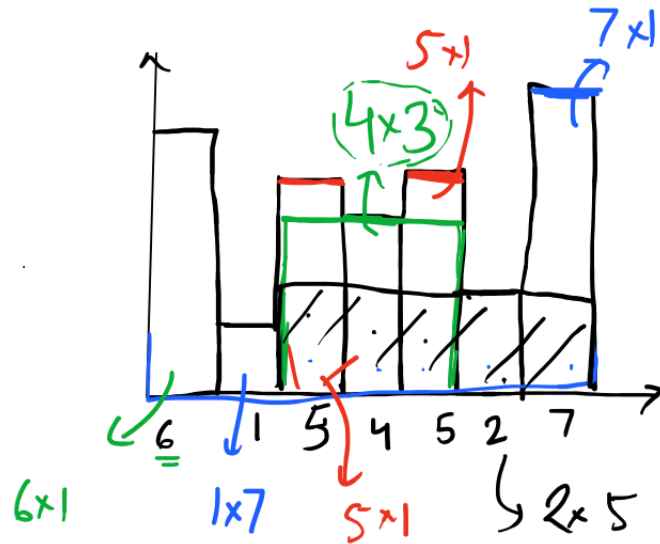
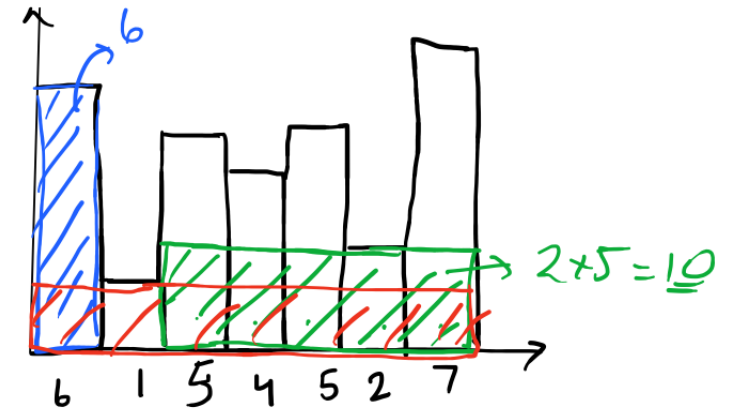
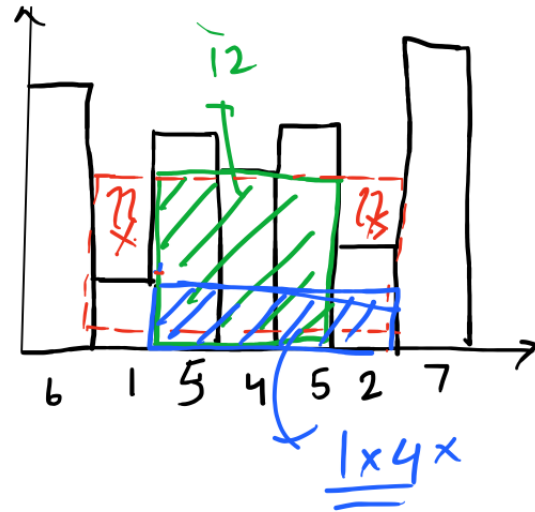


{ 1, 1, 1, 2, 1, 4, 7 }

* Largest Area Histogram \rightarrow L.C Hard



int[] hist
= { 6, 1, 5, 4, 5, 2, 7 }



1) Brute Force

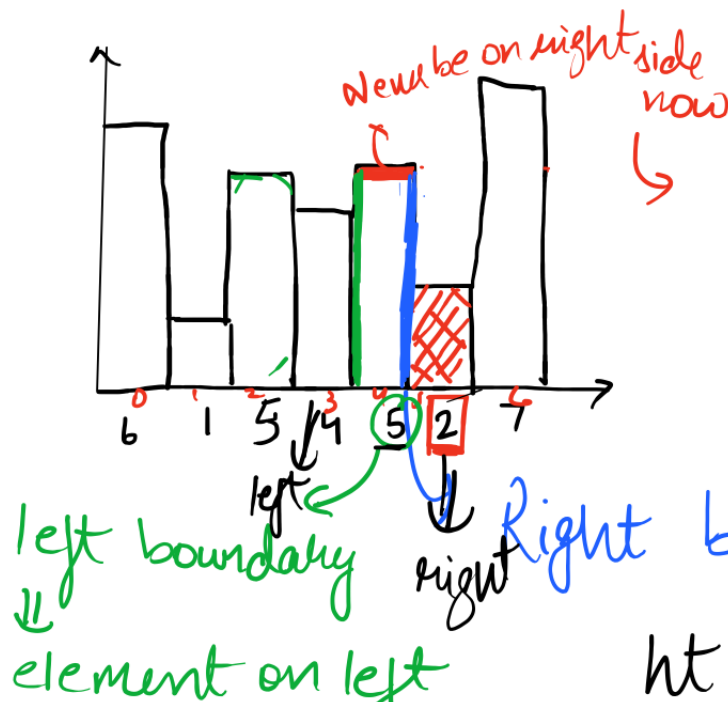
For every index

↳ consider current bar as the maximum and count elements before + after that can be included

T, .

for () {
 for () {
 for () {
 }
 }
 }
 $O(N^2)$ ✓
 ~~$O(N^3)$~~

for (int i = 0; i < n; i++) {
 for (j = i - 1; j > 0; j--) count before
 for (k = i + 1; k < n; k++) count after
 calculate the result
 as $hist[i] \times width$



2) Using Stack to calculate
NSL

2 blocked the bar at index 4

Calculate the answer for
 the blocked bar

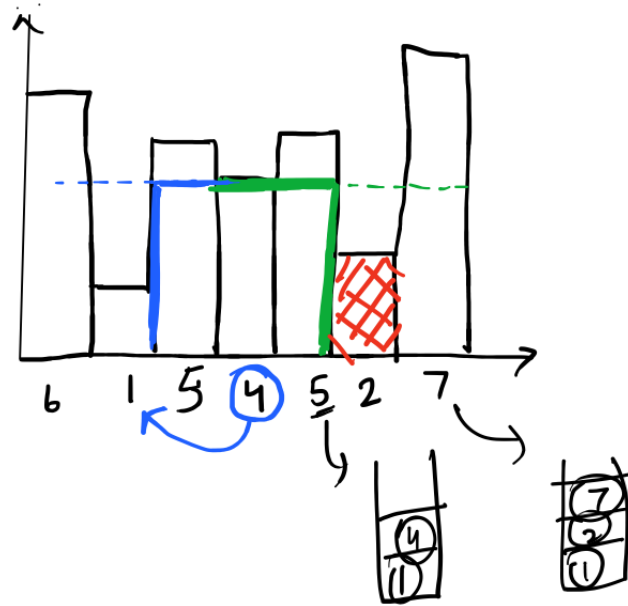
$$ht \times (right - left - 1)$$

just smaller
than the bar

$$\downarrow \quad \downarrow \quad \downarrow$$

$$5 \times 1 \quad 5 - 3 - 1 = 1$$

area



Consider i as the
NSR for elements
present in Stack

~~6~~ 12 ✓

Ans

Closest Smaller element
on left

N.S.R

Indexes of N.S.L & N.S.R

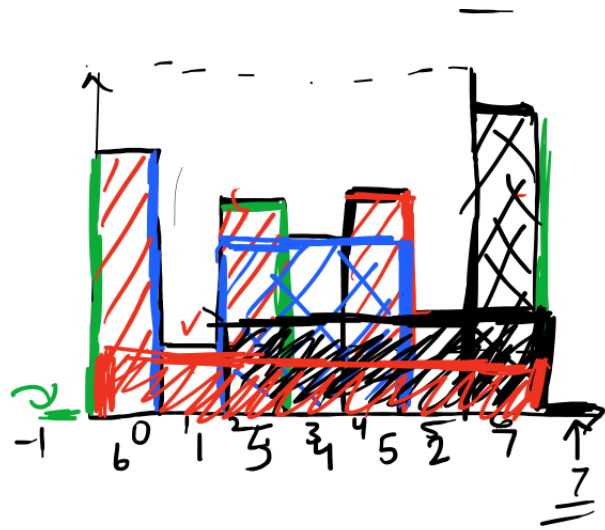
N.S.L + N.S.R

left

Right

Single Stack to calculate

N.S.R



$$\underline{\underline{1 = -10}}$$

$$NSR = n$$

$$w\bar{x} = 1$$

$$NSL = -1$$

$$Area = (1 \times (n-1)-1) = 1 \times 7+1-1 = 7$$



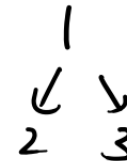
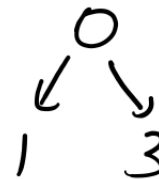
$$TC: O(n)$$

$$SC: O(n)$$

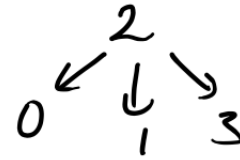
Celebrity Problem

	0	1	2	3
0	0	1	0	1
1	0	0	1	1
2	1	1	0	1
3	0	0	0	0

3x3



Celebrity
↳ who is known by all
by doesn't know anyone



③ ✓

$arr[i][j] = 0$
↳ i doesn't know j

$arr[i][j] = 1$
↳ i knows j
but doesn't mean $arr[j][i] = 1$??

Celebrity
↳ Row → All 0s
Col → All 1s (Except Self)

1) Brute force

- ↳ consider \forall from $[0, n)$
 - ↳ find i which follows the above properties
 - ↳ check Row & Col for entry i

$$TC: O(N^2)$$

$$SC: O(1)$$

2) Using Elimination

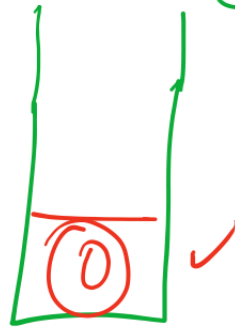
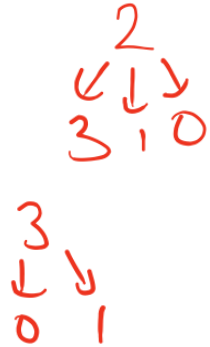
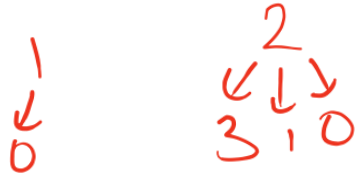
	0	1	2	3
0	0	1	0	1
1	0	0	1	1
2	1	1	0	1
3	0	0	0	0

↳ 0, ~~1~~, ~~2~~, 3

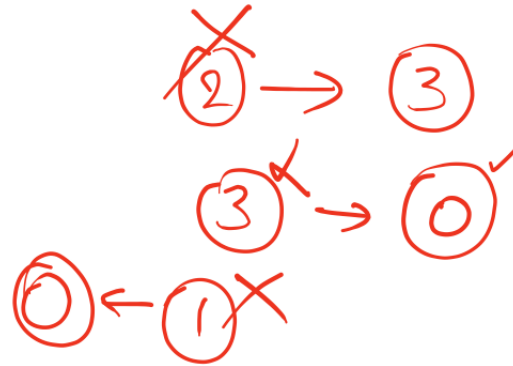
✓
① → 2 ✓
2 → 1

3
2
1
0

Pick 2 elements and



eliminate the non celebrity!



TC: $O(n)$
SC: $O(n)$