Next Greater Element on Right

Approach 2 :

nger

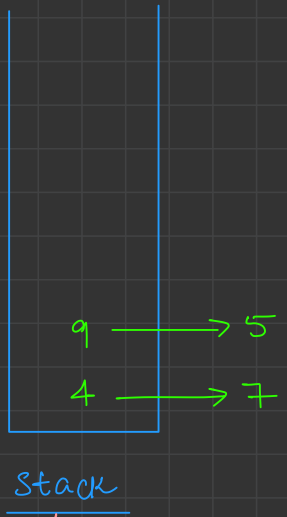
	6	7	2	7	-1	4	5	2	5	-1
arr[] :	3,	6,	1,	2,	7,	3,	4,	1,	2,	5

{ people looking for there nger

	6	7	2	7	-1	4	5	2	5	-1
	0	1	2	3	4	5	6	7	8	9
arr[] =	{3,	6,	1,	2,	7,	3,	4,	1,	2,	5};

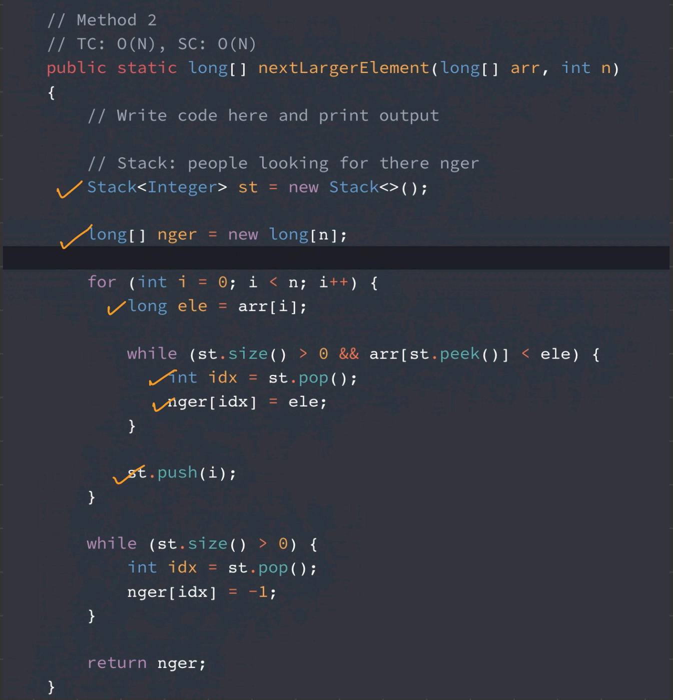
NOTE

① use index to store pos in stack.



↳ people looking for there nger

Monotic Stack



```java
// Method 2
// TC: O(N), SC: O(N)
public static long[] nextLargerElement(long[] arr, int n)
{
    // Write code here and print output

    // Stack: people looking for there nger
    Stack<Integer> st = new Stack<>();

    long[] nger = new long[n];

    for (int i = 0; i < n; i++) {
        long ele = arr[i];

        while (st.size() > 0 && arr[st.peek()] < ele) {
            int idx = st.pop();
            nger[idx] = ele;
        }

        st.push(i);
    }

    while (st.size() > 0) {
        int idx = st.pop();
        nger[idx] = -1;
    }

    return nger;
}
```

	6	7	2	7	-1	4	5	2	5	-1
	0	1	2	3	4	5	6	7	8	9
arr[] : {	3,	6,	1,	2,	7,	3,	4,	1,	2,	5 } 9
	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	-1	-1	6	6	-1	7	7	4	4	7

↓

ngel

Stack .

Monotic Stack !

monotic decreasing stack

Adv

① we use indexing here

② we can calc. other side greater / smaller elem. here.

Stock Span Problem

$$\text{price}[\,] = \left\{ \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 100, & 80, & 60, & 70, & 60, & 75, & 85 \end{array} \right\}$$

$$\{1,\ 1,\ 1,\ 2,\ 1,\ 4,\ 6\} \rightarrow \underline{\text{span}}$$

Bruteforce go to each day and try making longer span.

↳ TC: $O(N^2)$

SC: $O(1)$

$$\text{price}[\,] = \left\{ \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 100, & 80, & 60, & 70, & 60, & 75, & 85 \end{array} \right\}$$

✓ $$\text{ngeli} = \left\{ \begin{array}{ccccccc} -1, & 0, & 1, & 1, & 3, & 1, & 0 \\ 1 & 1 & 1 & 2 & 1 & 4 & 6 \end{array} \right\}$$

↓

next greater element on left's index

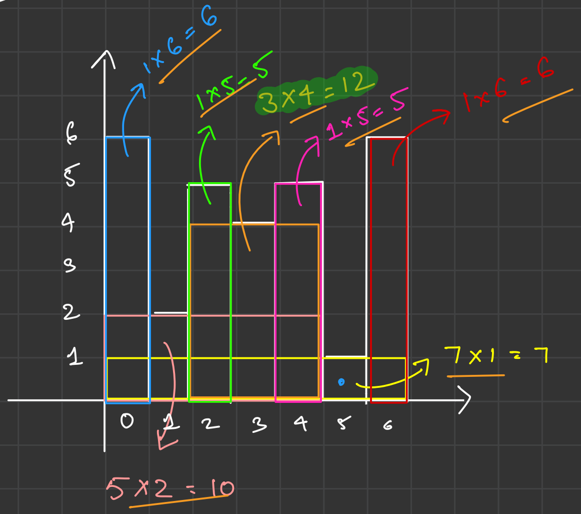
✓ $$\boxed{\text{span} = \text{index} - \text{ngeli}}$$

↓

* people looking for ngel.

+ move right to left

largest Area Histogram



$$\begin{array}{lllllll} 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{array}$$

heights[] = {6, 2, 5, 4, 5, 1, 6}

nseri[] = {1, 5, 3, 5, 5, 7, 7}

nseli[] = {-1, -1, 1, 1, 3, -1, 5}

1 5 1

width = r - l - 1

Steps

① get nsel i

② get nser i

③ Calc. width (r - l - 1)

④ Store max. area

Celebrity Problem

n → people in a gathering

$\boxed{n \times n} \rightarrow arr$

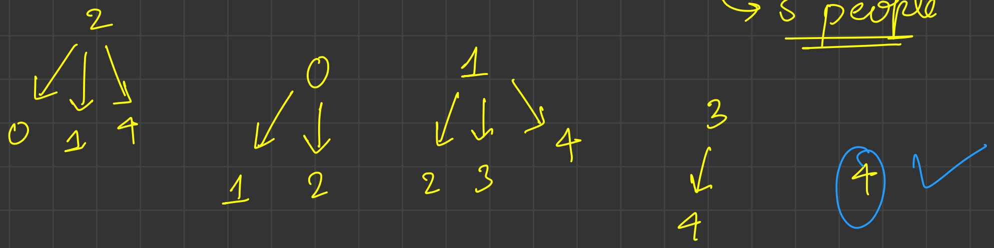
$$arr[\,][\,] : \begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & & \checkmark & \checkmark & X & X \\ \checkmark\ 1 & X & & \checkmark & \checkmark & \checkmark \\ 2 & \checkmark & \checkmark & & X & \checkmark \\ 3 & X & X & X & & \checkmark \\ 4 & X & X & X & X & \end{array}$$

5×5 → 5 people

Celebrity

✓ * who knows no one

* is known by everyone



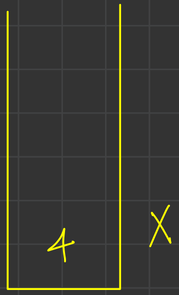
Brute force

for each person, all crosses in the row
and tick in the col

TC: $O(N^2)$
SC: $O(1)$

arr[][] :

$$\begin{array}{c|ccccc|c}
 & 0 & 1 & 2 & 3 & 4 & X \\
\hline
0 & & ✓ & ✓ & X & \boxed{X} & \\
1 & X & & ✓ & ✓ & ✓ & \\
2 & ✓ & ✓ & & X & ✓ & \\
3 & X & X & X & & ✓ & \\
4 & X & X & X & X & & \\
\hline
 & & & ✓ & & &
\end{array}$$



↳ potential celeb

potential celebs ⇝ 0, 1, 2, 3, 4

Elemination

O(N) + O(N)

↳ verify as celeb

getting any two potential Celeb,

Suppose $p1$ & $p2$

if $p1 \xrightarrow{\text{knows}} p2$

$p1$ can't be a celeb

else $p1 \xrightarrow{\text{doesn't know}} p2$

$p2$ can't be a celeb,

verify last potential Celeb as a actual Celeb or not.