

Next Greater Element to the right:

array →
strictly greater.

0	1	2	3	4	5	6	7	8
2	5	9	3	1	12	6	8	7

✓ ✓
↔

✓

✓

✓

✗

✓

✗

✗

↑

5

9

12

12

12

1

8

1

1

main order of values
in stack as decreasing
from Bottom to top.

Order - Maintenance

Souvik Dey
→ Deloitte Digital
→ Public Cts Student
→ Principle Global
sol"

Aakash Tyagi
→ Hushed In.
Anmol Girdhar
Kritika
①
②
③

Himanshu
→ Morgan Stanley
Anshul Gang.
→ VMware
→ Morgan Stanley

Mithun
→ Oracle
→ Goldman Sachs
→ Myndra.
Farie:



8-7

7-8

6-6

5-12

4-1

3-3

2-9

1-5

Next greater - II

$i > \text{arr.length}$
helper zone

virtual Double Array -

0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2	7	4	6	2	3	1	2	7	4	6	2	3

13 % 7 = 6

first half

problem
[$i < \text{arr.length}$]

Helper to solve first half

↓
in actual implementation

Do not push
in stack
because they
are just helper -

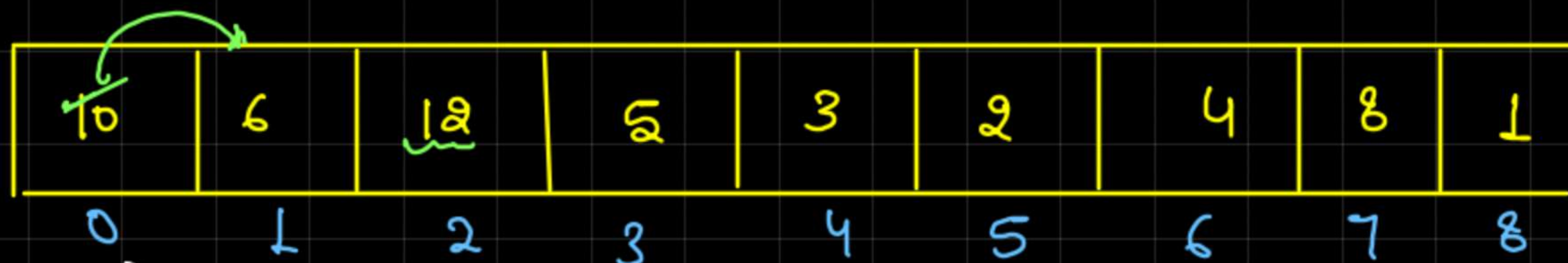
for ($i=0$; $i < 2 * \text{arr.length}$; $i++$) {

$\text{index} = i \% \text{arr.length};$

0	1	2	3	4	5	6
2	7	-1	6	7	3	7

~~8-1~~
~~7-8~~
~~6-4~~
~~5-2~~
~~4-3~~
~~3-5~~
~~2-12~~
~~1-6~~
~~0-10~~

8-1
 7-8
 6-4
 5-2
 4-3
 3-5
 2-12
 1-6
 0-10



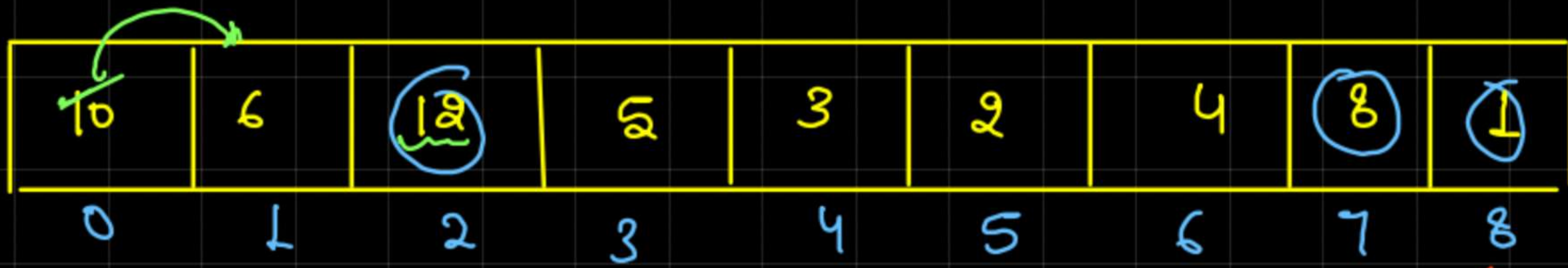
red } 12 12 -1 8 4 4 8 -1 -1
 push → Index

Result

pop → while

① st.size() > 0 & st.peek() < currElemt

what i am doing -
 ↳ Maintain order
 of value in
 decreasing order
 for "next greater
 Element to the right".



res →

2 2 7 6 6 7 -1 ↑

arr.length

~~8~~
~~7~~
~~6~~
~~5~~
~~4~~
~~3~~
~~2~~
~~1~~
~~0~~

```
private static int[] ngr(int[] arr) {
    Stack<Integer> st = new Stack<>();
    // push index in stack
    int[] res = new int[arr.length];
    st.push(0);
    for(int i = 1; i < arr.length; i++) {
        while(st.size() > 0 && arr[st.peek()] < arr[i]) {
            res[st.pop()] = i;
        }
        st.push(i);
    }
    while(st.size() > 0) {
        res[st.pop()] = -1;
    }
    return res;
}
```

if value is required
put arr[i]
rather than 'i'

Required result

Next greater Element to the left

Note: Check from Right to left

Next smaller Element to the right:

Home work

Daily temperature
Stock span.

10	6	12	15	3	2	4	8	7
0	1	2	3	4	5	6	7	8

res →

6 3 3 3 2 1 1 7 ↓



order of stack's value
Increasing order from bottom to top