Next greater element I:-

arr

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

Brute Force:-

For any element arr[i], search from i+1 to n-1. The first element which is greater than arr[i] will be the answer.

T = O(n^2) => We have to find answer for each i (0 to n-1)

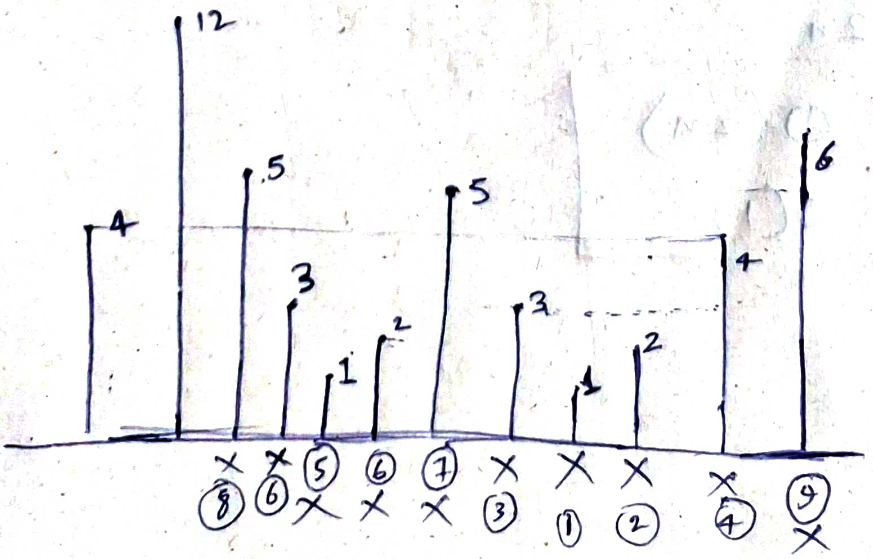
S = O(n)

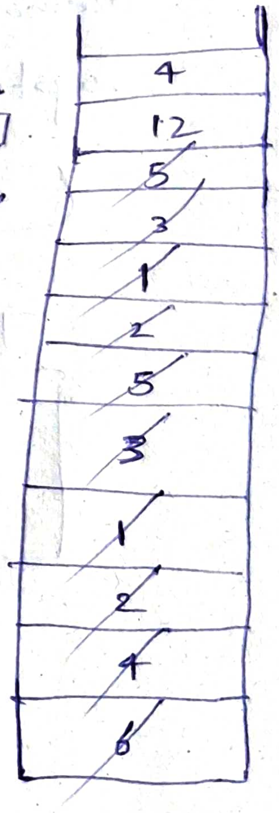
↳ Because we will be storing output in a separate array. If you store output in the same array then S = O(1)

Optimal Approach

←

	4	12	5	3	1	2	5	3	1	2	4	6
	0	1	2	3	4	5	6	7	8	9	10	11
~~i~~	i	~~i~~	~~i~~	~~i~~	~~i~~	~~i~~	~~i~~	~~i~~	i	i	~~i~~	~~i~~
	12	-1	6	5	2	5	6	4	2	4	6	-1





Stack (Monotonic)

Time complexity is littical tricky to understend.

Think logically

For each element in the array we are not removing all the elements present in stack.

If for one element arr[i] we remove some elements from stack. Now, for some ~~elements~~ ~~some~~ other element arr[j] previously removed elements will not be removed again.

So, for arr[i] we removed 2 elements. For arr[i-1] we remove 3 elements. And so on upto $i=0$.

$$\boxed{2+3+-----=n}$$

$T=2n$

$$\boxed{\begin{array}{l} T = O(2n) \\ S = O(1) \end{array}}$$