

32

```
public static int[] solve(int[] arr) {
    // Your Code Here
    int n = arr.length;
    int[] res = new int[n];
    Stack<Integer> st = new Stack<>();
    for(int i = 0; i < n; i++){
        while(st.size() > 0 && arr[st.peek()] <= arr[i]) st.pop();
        if(st.size() == 0){
            res[i] = -1;
        } else res[i] = st.peek();
        st.push(i);
    }
    return res;
}
```

[1 2 2 1 3 3]

i=0 i=1 i=2 i=3 i=4
0 0 2 1

5

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

while (st.size() > 0 && arr[st.peek()] <= arr[i])
st.pop();

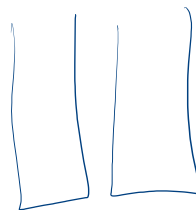
if (st.size() == 0) {
res[i] = -1;
} else {
res[i] = st.peek();
} st.push(i);

(aa(bdc))p(dee) ∈
→ 1 2 2 1 3 3

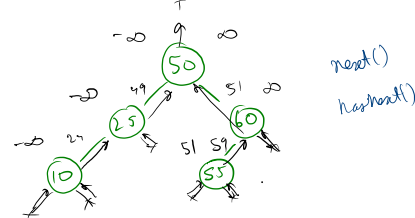
next = 4

next = 1
for (i = 0; i < s.length(); i++)
C = s.charAt(i)
if (C == '(') {
res.add(next)
st.push(next)
next++
} else if (C == ')') {
res.add(st.pop())
}
return res;

10



10 25 50 55 60



next()
hasNext()