Todays Content

Sum of all subarrays max

Q Given ar[N] sum of max of all subarrays

Ex1:

	0	1	2
ar[ ] = {	2	5	3 }

#Subarrays	#Max A = 25
{2}	2
{2 5}	5
{2 5 3}	5
{5}	5
{5 3}	5
{3}	3

#Q1 : launch in class

#Idea1

for every subarray:
Iterate & calc max & add in ans.

TC: $O(N^2 * N) = O(N^3)$ SC: $O(1)$

#Idea2

	0	1	2	3	4	5
ar[] = {	3	2	4	5	7	6 }

max = 3 + 3 + 4 + 5 + 7 + 7 = ?

for every subarray:
calc max using carry forward & add in ans

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

#Idea3: Contribution Technique:

Add contribution of each ar[i]

Contribution of ar[i] = ar[i] * {#No: of subarrays in which ar[i] is max}

	0	1	2
ar[] = {	2	5	3 } ans:

#Subarrays	#Max
{2}	2
{2 5}	5
{2 5 3}	5
{5}	5
{5 3}	5
{3}	3

2*1 + 5*4 + 3*1 = 25

Q Calculate in how many subarrays else arr[5] = 9 is max?

DryRun1:

	0	1	2	3	4	5	6	7	8	9	10
arr[] = {	2	3	13	7	6	9	5	3	2	11	2 }

#Subarray:

s	e	#Subarray = 12
3	5	{3 5} {3 6} {3 7} {3 8}
4	6	{4 5} {4 6} {4 7} {4 8}
5	7	{5 5} {5 6} {5 7} {5 8}
	8	

DryRun2:

	0	1	*P1 2	3	4	i 5	6	7	8	*P2 9	10
arr[] = {	-2	-3	-13	-7	-6	9	-5	-3	-2	-11	-2 }

#start $(P_1 .. i]$ #end $= [i .. P_2)$

$s = i - P_1 = 3$ $e = P_2 - i = 4$

#Subarray $= s * e = 12$

#P_1 = i'th greater index in left for arr[i] # Default : -1

#P_2 = i'th greater index in right for arr[i] # Default = N

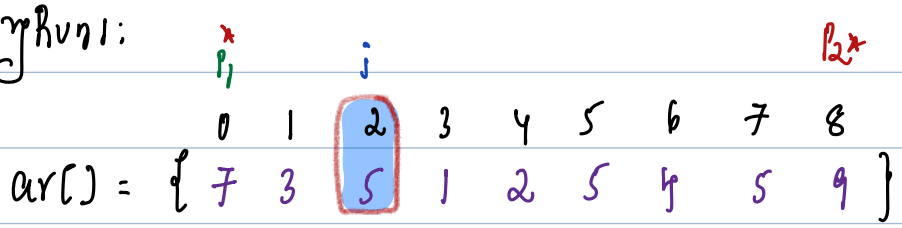
$[a\ b] = \#b - a + 1$

$(a\ b] = \#b - a$

$[a\ b) = \#b - a$

Q Find in how many subarrays ele ar[2]=5 is max, based on above approach.

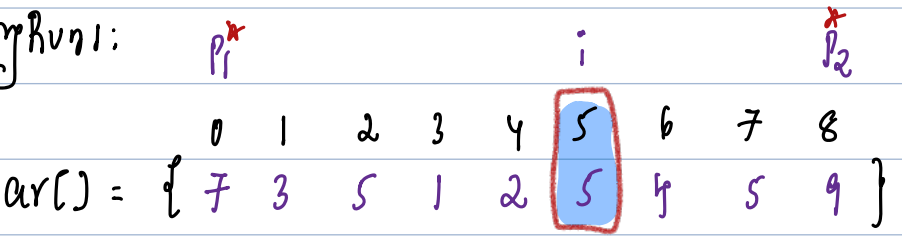
DryRun1:



#start [p1..i]	#end = [i.. p2)	# subarrays
1	2	{1 2} {1 3 3} {1 4} {1 5} {1 6 3} {1 7}
2	3	{2 2} {2 3 3} {2 4} {2 5} {2 6 3} {2 7}
	4	
	5	
	6	
	7	

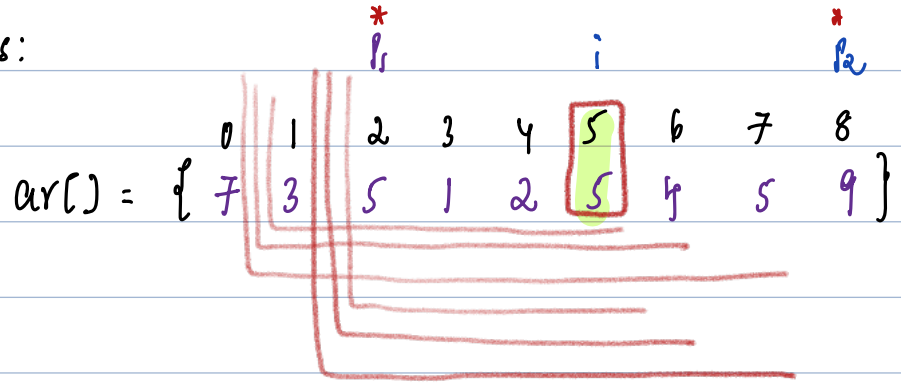
Q Find in how many subarrays ele ar[5]=5 is max, based on above approach.

DryRun1:



#start [p1..i]	#end = [i.. p2)	
1	5	# {1 5} {1 6} {1 7}
2	6	# {2 5} {2 6} {2 7}
3	7	# {3 5} {3 6} {3 7}
4		# {4 5} {4 6} {4 7}
5		# {5 5} {5 6} {5 7}

#obs:



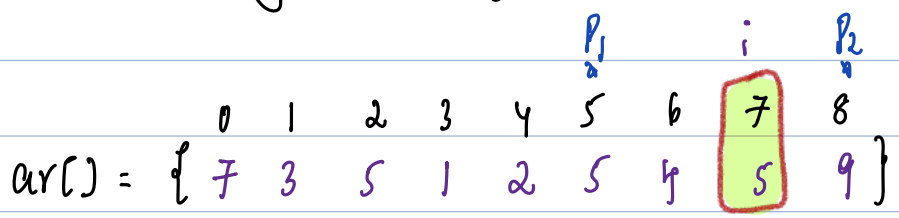
#obs1:

P_1 = 1st greater or equal index on left for ar[i]

P_2 = 1st greater index on right for ar[i].

#start $[P_1 .. i]$	#end = $[i .. P_2)$	
3	5	# {3 5} {3 6} {3 7}
4	6	# {4 5} {4 6} {4 7}
5	7	# {5 5} {5 6} {5 7}

Q find in how many subarrays ele ar[7] = 5 is max, based on above approach



#start $[P_1 .. i]$	#end = $[i .. P_2)$	
6	7	= {6 7} {7 7}
7		

#Conc: P_1 = 1st greater or equal index on left for ar[i]

P_2 1st greater index on right for ar[i].

or

P_1 = 1st greater index on left for ar[i]

P_2 1st greater or equal index on right for ar[i].

```
int maxSub(vector<int> &ar){
    int N = ar.size();
    # Nearest greater or equal ele in left.
        stack<int> st;
        vector<int> P1(N,-1);
        for(int i=0; i<N; i++){
            while(st.size()>0 && ar[st.top()]<=ar[i])
                st.pop();
            if(st.size()>0){
                ar[i] = st.top();
            }
            st.push(i);
        }
    # Nearest greater ele in right
        stack<int> st;
        vector<int> P2(N,N);
        for(int i=0; i<N; i++){
            while(st.size()>0 && ar[st.top()]<=ar[i])
                st.pop();

            if(st.size()>0){
                ar[i] = st.top();
            st.push(i);
        }

    int A=0;
    for(int i=0; i<N; i++){
        A = A + ar[i]*(i-P1[i])*(P2[i]-i)
    }
}
return A;
```