

## 6.6 SUM OF SUBARRAY MINIMUMS

### Question:

Given an array of integers arr, find the sum of min(b), where b ranges over every (contiguous) subarray of arr. Since the answer may be large, return the answer modulo  $10^9 + 7$ .

### AIM

To compute the sum of subarray minimums efficiently using a monotonic stack approach in Python.

### ALGORITHM

1. Initialize modulo =  $10^9 + 7$ .
2. Use a monotonic stack to find Previous Less Element (PLE) and Next Less Element (NLE) for each element.
3. PLE[i] gives distance to the previous smaller element of arr[i].
4. NLE[i] gives distance to the next smaller element of arr[i].
5. The contribution of arr[i] is arr[i] \* PLE[i] \* NLE[i].
6. Sum up contributions of all elements and return modulo  $10^9 + 7$ .

### PROGRAM

```
def sum_subarray_mins(arr):
    mod = 10**9 + 7
    n = len(arr)
    stack = []
    prev = [0] * n
    next_ = [0] * n

    for i in range(n):
        while stack and arr[stack[-1]] > arr[i]:
            stack.pop()
        prev[i] = i - stack[-1] if stack else i + 1
        stack.append(i)

    stack = []
    for i in range(n - 1, -1, -1):
        while stack and arr[stack[-1]] >= arr[i]:
            stack.pop()
        next_[i] = stack[-1] - i if stack else n - i
        stack.append(i)

    total = sum(arr[i] * prev[i] * next_[i] for i in range(n))
    return total % mod

arr = list(map(int, input("Enter array: ").split()))
print("Sum of subarray minimums:", sum_subarray_mins(arr))
```

Input:

arr = [3,1,2,4]

Output:

```
>>> Enter array: 3 1 2 4
Sum of subarray minimums: 17
>>> |
```

### **RESULT:**

Thus, the program is successfully executed and verified to compute the sum of subarray minimums.

### **PERFORMANCE ANALYSIS:**

Time Complexity:  $O(n)$ , where  $n$  is the number of elements in the array, due to monotonic stack usage.

Space Complexity:  $O(n)$  for storing stack, PLE, and NLE arrays.