Contribution Technique Approach

In class / Sum of all subarray sums

(Introduction to Problem Solving - I)

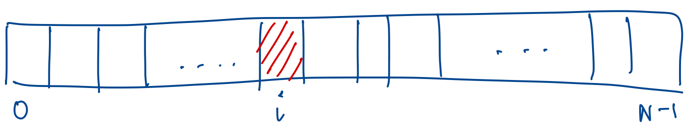
- How many times an element appears in all the subarrays?

arr → [3 -2 4 -1 2 6]

0 1 2 3 4 5

⇒ suppose we need to find out how many times a particular index occurs in all the subarrays.

⇒ Let us suppose there are 'N' elements in total & that particular element is present at the 'i' th index.



Observation

→ For elements at index '0' till 'i-1', for each of the elements in that range, our element at index 'i' will be a part of all the Subarrays which go upto the extent of 'i' or till 'N-1'

→ For example for element at index '0', element at index 'i' will be a part of the following subarrays

[0, 1, ..., i] , [0, 1, ..., i, i+1] , ... , [0, 1, ..., i, i+1, i+2, ... N-1]

→ Similarly for element at index 'i', element at index 'i' will be a part of the following subarrays

$$\underbrace{[1, 2, \ldots i] , [1, 2, \ldots, i, i+1] , \ldots , [1, 2, \ldots, i, i+1, \ldots, N-1]}$$

Basically elements within index range [i, N-1]

= N-1 -i + 1 = (N-i)

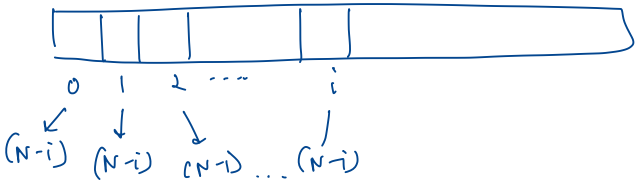
⇒ for each element before the element at index 'i', element at index 'i' appears exactly (N-i) times in all it's subarrays

⇒ Also, element at index 'i' also appears exactly (N-i) times in all it's subarrays

$$\underbrace{[i], [i, i+1], [i, i+1, i+2], \ldots, [i, i+1, i+2, \ldots, N-1]}$$

range = [i, N-1] = N-1-i+1 = (N-i)

⇒ for elements with index > 'i', element at index 'i' will never be a part of any of its Subarrays

⇒ 

$(N-i) + (N-i) \ldots$ $(i+1)$ times $=$ $(N-i)(i+1)$

⇒ frequency of element at index 'i' in all the subarrays of the original array $= (N-i)(i+1)$

∴ In the sum of all the subarrays of the array, contribution of the an element in the sum $=$ $arr[i] \times (N-i) \times (i+1)$

Sum of all subarray sums

```
int sum = 0;
for ( int i = 0; i < N; i++) {
|     sum ± arr[i] × (N-i) × (i+1)
}
```

T.C. - O(N)
S.C. - O(1)