Arrays – Subarrays

1. Continuous part of an array is called Subarray.
2. A single element is a Subarray.
3. Entire array is a Subarray.
4. Empty cannot be a Subarray.

$ar[a]$ =

0	1	2	3	4	5	6	7
4	5	1	9	0	2	3	5

Quiz 1

a) [5]

b) [4, 5, 1, 0] ✗

c) [9, 0, 2, 3]

d) [4, 5, 1]

a, c, d

To uniquely define a subarray –

subarray [start   end]

subarray [i   j] → length = $j - i + 1$

$(i <= j)$

Count of subarrays

	0	1	2	3	4	5	6
[	4	2	10	3	12	-2	15]

Subarrays starting from 0th index:

7

Subarrays starting from 1st index: **Quiz 2**

6

Quiz 3

start	count
0	7
1	6
2	5
3	4
4	3
5	2
6	1
	28

$$\frac{7(7+1)}{2} = \frac{7 \times 8^4}{2}$$

$$= 28$$

Given N array elements, how many subarrays can be generated ?

$$ar[N] = \begin{bmatrix} 0 & 1 & 2 & 3 \ldots\ldots & i & i+1 & \ldots\ldots & N-2 & N-1 \end{bmatrix}$$

Start 0

$$[0 \quad 0]$$

$$[0 \quad 1]$$

$$[0 \quad 2]$$

$$\vdots$$

$$[0 \quad N-1]$$

$$\underline{\underline{N}}$$

Start 1

$$[1 \quad 1]$$

$$[1 \quad 2]$$

$$[1 \quad 3]$$

$$\vdots$$

$$[1 \quad N-1]$$

$$\underline{\underline{N-1}}$$

Start 2

$$[2 \quad 2]$$

$$[2 \quad 3]$$

$$[2 \quad 4]$$

$$\vdots$$

$$[2 \quad N-1]$$

$$\underline{\underline{N-2}}$$

$$\ldots \sim$$

$$\underline{\underline{2}}$$

$$\underline{\underline{1}}$$

Start N-2

$$[N-2 \quad N-2]$$

$$[N-2 \quad N-1]$$

Start N-1

$$[N-1 \quad N-1]$$

Total = $N + N-1 + N-2 + N-3 + \ldots\ldots 2 + 1$

$$= \frac{N(N+1)}{2}$$

Q1. Print all values of a subarray

array
↓
start
↓
end
↙

```
printSubarray(A[ ], s, e) {

      for ( i = s ;  i <= e; i++ ) {

              print ( A [i] )


      }



}
```

Time - O(N)
Space - O(1)

Q2. Find the sum of all elements in a given subarray.

```
addSubarray(A[ ], s, e) {

      sum = 0
      for ( i = s ;  i <= e; i++ ) {

              sum += A [i]


      }

      return   sum

}
```

Time - O(N)
Space - O(1)

Q3. Print all subarrays of a given array.

A: | 0 | 1 | 2 |

A: 2 8 9

N=4 [0 1 2 3]

[0 0]   2   [0   0]

[0 1]   2 8   [0   1]   [1   1]

[0 2]   2 8 9   [0   2]   [1   2] [2 2]

[1 1]   8   [0   3]   [1   3] [2 3] [3 3]

[1 2]   8 9

[2 2]   9

$i \rightarrow [0, N-1]$

$j \rightarrow [i, N-1]$

for (i=0; i<n; i++) {   ← N

    for (j=i; j<n; j++) {   ← N

      // Subarray [i   j]

      print Subarray(A, i, j)   ← N

    }

}

Quiz 4

No of subarrays = $\frac{N(N+1)}{2}$ ≈ N^2

Time to print 1 subarray — N

TC: $O(N^3)$

SC: $O(1)$

Q4. Print sum of every single subarray.

$$A = \begin{array}{ccc} {}^{0} & {}^{1} & {}^{2} \\ 3 & -2 & 4 \end{array}$$

[0 0]   3

[0 1]   1

[0 2]   5

[1 1]   -2

[1 2]   2

[2 2]   4

```
SumOfSubarrays(int []A) {
```
        for (i = 0; i < n; i++) {   ← N

                for (j = i; j < n; j++) {   ← N
                        // Subarray [i   j]
                        s = add Subarray (A, i, j)   ← N
                        print (s)
                }

        }

```
}
```

Time – $O(N^3)$

Space – $O(1)$

Optimisation

$$addSubarray(A, i, j) \rightarrow sum[i \quad j]$$

$$sum[L \quad R] = \begin{cases} \text{if } L==0: & pf[R] \\ \\ \text{else}: & pf[R] - pf[L-1] \end{cases}$$

```
SumOfSubarrays(int []A) {
        // Construct the Pf array first    ← TODO
        for (i=0; i<n; i++) {    ← N        ← O(N) time

                for (j=i; j<n; j++) {    ← N
                        // sum [i   j]
                        if ( i==0)
                                s= pf[j]
                        else
                                s = pf[j] - pf[i-1]

                        print (s)

                }

        }

}
```

Quiz 5

Time - $O(N^2)$
Space - $O(N)$

Expected
SC: O(1)

$$ar[7] = \begin{bmatrix} 7 & 3 & 2 & -1 & 6 & 8 & 2 & 5 \end{bmatrix}$$

indices: 0 1 2 3 4 5 6 7

[2 2]   2

[2 3]   1

[2 4]   7

[2 5]   15

[2 6]   17

[2 7]   22

for (j=2; j<n; j++) {

    s = addSubarray (A, 2, j)

    print (s)

}

TC: O(N²)

$$ar[7] = \begin{bmatrix} \overset{0}{7} & \overset{1}{3} & \overset{2}{2} & \overset{3}{-1} & \overset{4}{6} & \overset{5}{8} & \overset{6}{2} & \overset{7}{5} \end{bmatrix}$$

[2 2]   2 + -1

[2 3]   1 +6

[2 4]   7 +8

[2 5]   15 + 2

[2 6]   17 +5

[2 7]   22

Carry forward → Sum of previous subarray

from L to R

Sum = 0

for (j = 2 ; j < n ; j++) {

   Sum + = A[j]

   print (sum)

}

TC : O(N)

SC : O(1)

Q6 **Print all subarray sum starting at index = 3**

```
Sum = 0

for ( j = 3 ; j < n ; j++ ) {
        sum  + = A[j]

        print (sum)

}
```

Q7 **Print all subarray sum starting at index = i**

i   is   given in   input

```
Sum = 0

for ( j = i ; j < n ; j++ ) {
        sum  + = A[j]

        print (sum)

}
```

Q8 What will this code do ?

```
for ( i=0; i < N; i++ ) {
    Sum = 0
    for ( j = i; j < n; i++ ) {
        Sum += A[j]
        print (sum)
    }
}
```

Print all subarray sums
using carry forward

Time – $O(N^2)$

Space – $O(1)$

Break till 10:25 PM

         0   1   2
A :   3   -1   4

s	e	sum
[0	0]	3
[0	1]	2
[0	2]	6
[1	1]	-1
[1	2]	3
[2	2]	4
		___
		17

```
total = 0
for (i=0; i<N; i++) {
    sum = 0
    for (j=i; j<n; j++) {
        sum += A[j]
        total += sum
    }
}
print ( total )
```
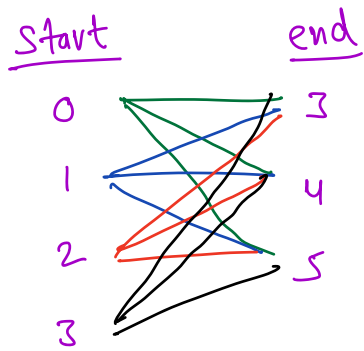
Time - O(N^2)
Space - O(1)

         0   1   2
    A : 3   -1   4

s	e	sum	
[0	0]	3	3
[0	1]	2	3 + -1
[0	2]	6	3 + -1 + 4
[1	1]	-1	-1
[1	2]	3	-1 + 4
[2	2]	4	4
		17	3 × 3 + -1 × 4 + 4 × 3

If we know in
how many subarrays
each element is
coming, can we
solve this
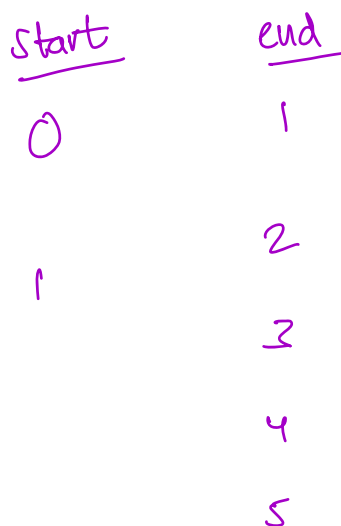faster ?

Q10. In how many subarrays index 3 is present ?

	0	1	2	3	4	5
A =	3	-2	4	-1	2	6

start   end

start	end
0	3
1	4
2	5
3	

$$\underline{4} \times \underline{3} = 12$$

Q11. In how many subarrays index 1 is present ?

	0	1	2	3	4	5
A =	3	-2	4	-1	2	6

start	end
0	1
	2
1	3
	4
	5

$$\underline{2} \times \underline{5} = 10 \quad \text{subarrays}$$

Q12. Given arr[N], in how many subarrays index i is present ?

ending indexes

$A = \quad a_0 \quad a_1 \quad a_2 \quad a_3 \quad \ldots, \quad a_{i-1} \quad a_i \quad a_{i+1} \quad \ldots \quad a_{n-1}$

starting indexes

Start : $\begin{bmatrix} 0 & & i \end{bmatrix} \Rightarrow (i+1)$ elements

end : $\begin{bmatrix} i & & N-1 \end{bmatrix} \Rightarrow (N-i)$ elements
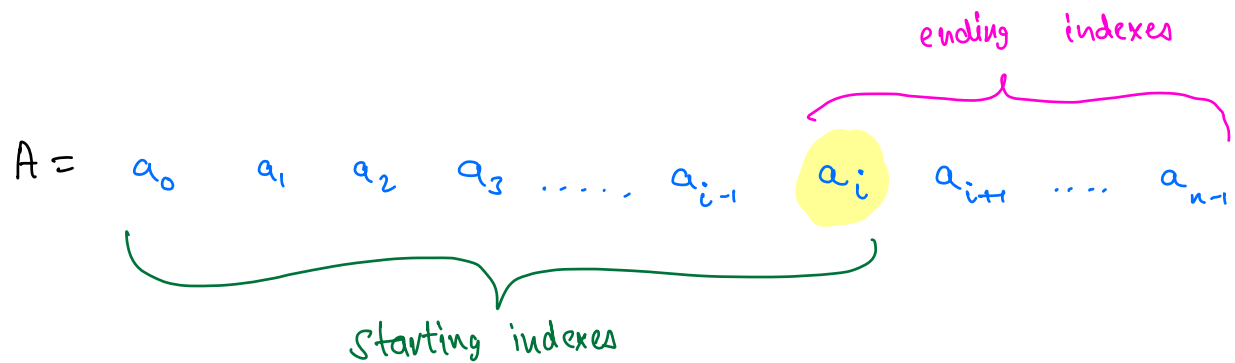
In how many subarrays is i^{th} element present

$\rightarrow (i+1) \times (N-i)$

Back to Q9

A :   3   -1   4

N = 3

Contribution
Count
$(i+1) \times (N-i)$

       3   4   3

$i=0 \rightarrow (0+1) \times (3-0)$

$i=1 \rightarrow (1+1) \times (3-1)$
$= 2 \times 2$

$i=2 \rightarrow (2+1) \times (3-2)$
$= 3 \times 1$

final ans $= \boxed{3 \times 3} + \boxed{-1 \times 4} + \boxed{4 \times 3}$

$= 17$

A :   3   -1   4

s	e	sum		
[0	0]	3	3	
[0	1]	2	3 + -1	
[0	2]	6	3 + -1 + 4	
[1	1]	-1	-1	
[1	2]	3	-1 + 4	
[2	2]	4	4	

17    $3 \times 3 + -1 \times 4 + 4 \times 3$

Contribution
Technique

total = 0
for (int i=0; i<N; i++) {
     count = (i+1) × (N-i)
     total += A[i] × count
}
print (total)

TC: O(N)
SC: O(1)

Doubts

Pick from both sides

$A = 5, -2, 3, 1, 2$

$B = 3$

Pick 3 elements

$5, -2, 3, 1, 2$

$5 + -2 + 3 = 6$

$5 + -2 + 2 = 5$

$5 + 1 + 2 = 8$ ← Ans

$3 + 1 + 2 = 6$

$A = \quad 2, 3, -1, 4, 2, 1$

$B = \quad 4$

$2, 3, -1, 4, 2, 1$

$2 + 3 - 1 + 4 = 8$

$2 + 3 - 1 + 1 = 5$

$2 + 3 + 2 + 1 = 8$

$2 + 4 + 2 + 1 = 9$ ← Ave

$-1 + 4 + 2 + 1 = 5$

$2, 4, 8, 7, 6, 10$

even length

first & last element of each subarray has to be even

Good Night

Thank You

Monday