

Q Max sum out of all subarrays

Ideas:

[1, 2, 3, -4]

1) Take sum for all subarrays

```
for (s=0; s<n; s++) {
```

TC: $O(n^3)$

```
    for (e=s; e<n; e++) {
```

2: 12

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

2: 13

2) Carry Fwd

2: 12 + 13ⁿ

```
for (s=0; s<n; s++) {
```

```
    sum = 0
```

```
    for (e=s; e<n; e++) {
```

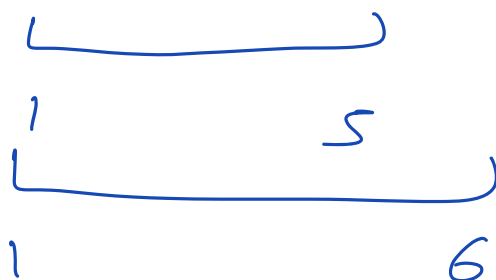
```
        sum += a[e]
```

TC: $O(n^2)$

```
        ans = max(ans, sum)
```

```
    }
```

```
}
```



Q Find sum of all subarray sum

Eg [1, 2, 3]

[1]	1
[1 2]	3
[1 2 3]	6
[2]	2
[2 3]	5
[3]	3
	<hr/>
	20
	<hr/>

Idea: Use above code of carry forward only

total = 0

for (s=0 ; s<n ; s++) {

sum = 0

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

sum += a[e]

total += sum

{ | }

Eg 0 1 2
 1 2 3

$$S=0$$

$$S=1$$

$$S=2$$

$$\begin{array}{r} 1 \quad 3 \\ 6 \quad 2 \\ 5 \quad 3 \\ \hline 20 \end{array}$$

Optimal:

[1, 2, 3]

1

1

1 2

1 + 2

1 2 3

1 + 2 + 3

2

2

2 3

2 + 3

3

3

$$= \underline{\underline{1 \times 3 + 2 \times 4 + 3 \times 3}}$$

$$= \underline{\underline{20}}$$

Count how many times a[i] occurs.

Quiz

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

$i=1$

$s = 2$ $[0, i]$

$e = 5$ $[i, n-1]$

total ans = $2 \times 5 = 10$

1 home 1
 s e

2 B

5 G

$i=2$

s $[0, i] = [0, 2] = 3$

e $[i, n-1] = [2, 5] = 4$

3×4

$= 12$

$$b - a + 1$$

Obs index $i \Rightarrow$ $n-1 - i + 1$
 $n-i$

$$0 \quad 1 \quad 2 \quad \dots \quad i \quad \dots \quad n-1$$

$$[0, i]$$

$$[i, n-1]$$

$$\text{contribution} = (i+1) \times (n-i)$$

$$\text{sum} = 0$$

$$\text{for } (i=0 ; i < n ; i++) \{$$

$$\text{contribution} = (i+1)(n-i)$$

$$\text{sum} += a[i] * \text{contribution}$$

}

$$\text{return sum}$$

$$TC: O(n)$$

$$SC: O(1)$$

$$0 \quad 1 \quad 2$$

$$1 \quad 2 \quad 3$$

$$1 \times 1 \times 3 \quad + \quad 2 \times 2 \times 2 \quad + \quad 3 \times 3 \times 1$$

$$3 \quad + \quad 8 \quad + \quad 9 \quad = \quad 20$$

K size subarrays in array of size N

$[5, 6, 2, 9, 11, -3, 5, 3, 8]$ $N = 9$

$K = 1$

9

$K = 2$

8

$K = 3$

7

\Rightarrow size $k \Rightarrow$ no of subarrays
 $= n - k + 1$

Quiz

$N = 7$

$K = 4$

$n - k + 1$

$= 7 - 4 + 1 = 4$

Q1 Max sum of a subarray of
len = k

eg-
0 1 2 3 4 5 6 7 8 9
-3, 4, -2, 5, 3, -2, 8, 2, -1, 4

k=5

start	end	sum
0	4	7
1	5	8
2	6	12
3	7	16
4	8	10
5	9	11

ans = 16

Brute: Use prefix sum array to calc
all subarray sums in $O(1)$
TC: $O(n)$ SC: $O(n)$

Reduce SC to $O(1)$

Idea: Sliding window

0 1 2 3 4 5 6 7 8 9
-3 4 -2 5 3 -2 8 2 -1 4

s	e	sum
0	4	$-3 + 4 + -2 + 5 + 3$
1	5	$4 + -2 + 5 + 3 + -2$
2	6	$-2 + 5 + 3 + -2 + 8$
⋮	⋮	

add $ar[e]$

remove $ar[s-1]$

from prev sum, sum of $s:e$
new sum = $sum - ar[s-1] + ar[e]$

Code

```
// first calc sum of first window
sum = 0
for (i = 0; i < k; i++) {
    sum += a[i]
}

ans = sum
s = 1          e = k
while (e < n) {
    sum = sum - a[s-1] + a[e]
    ans = max(ans, sum)
    s++        e++
}

return ans
```

TC: $O(n)$
SC: $O(1)$

0 1 2 3 4 5 6 7 8 9
 -3 4 -2 5 3 -2 8 2 -1 4

$k=5$

0 4

1 5

2 6

3 7

4 8

5 9

6 10

7

$$7 - a(0) + a(5) = 8$$

$$8 - a(1) + a(6) = 12$$

$$12 - a(2) + a(7) = 16$$

$$16 - a(3) + a(8) = 10$$

$$10 - a(4) + a(9) = 11$$

STOP!!

$$\text{ans} = 16$$

{done}

0 1 2 3

1 2 3 4

$$2 \times 3 = 6$$

1, 2

1, 2, 3

1, 2, 3, 4

2

2, 3

2, 3, 4

ans = sum

s = 0 e = k-1

while (e < n) {

 sum = bf(e) - bf(s-1)

 ans = max(ans, sum)

 s++ e++

}

return ans

TC: $O(n)$

SC: $O(1)$

s = 0

bf(e)

0 1 2 3 4

apple

0, 0

0, 1

0, 2

0, 3

0, 4

= 5

$(i+1)(n-i)$

$$(0+1)(5-0) = 1 \times 5 = 5$$

0 1 2 3 4 5

amazon

0 0 1 2 3 4 5

→ 6

2 2 3 4 5

→ 4
10

L

R



Suffix

10

20

30

60

50

30

