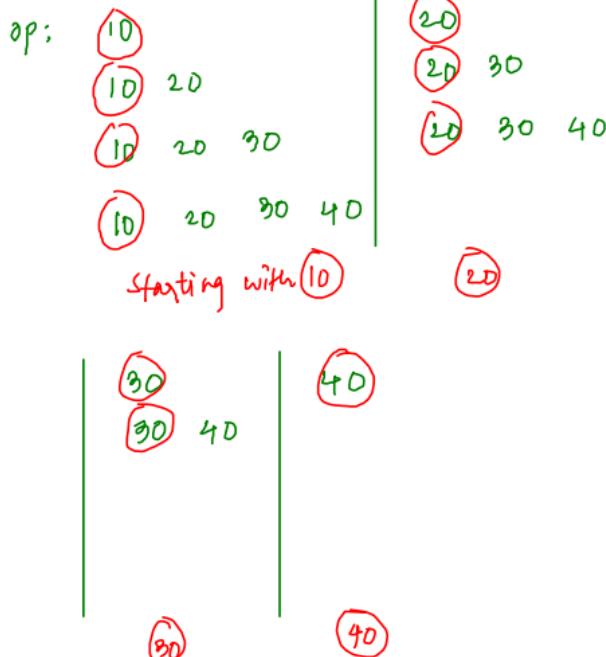


* Generate Subarray: (any continuous segment of the array)
 (order should be same)

e.g.: [10, 20, 30, 40]



Q: how to generate subarray's starting with 10?

$$\begin{aligned}
 & \text{* no. of iterations} \\
 & \text{end} = 0 \rightarrow \textcircled{1} \\
 & \text{end} = 1 \rightarrow \textcircled{2} \\
 & \text{end} = 2 \rightarrow \textcircled{3} \\
 & \text{end} = \textcircled{3} \rightarrow \textcircled{4} \\
 & \frac{1+2+3+4+\dots+n}{n(n+1)} \\
 & \approx \frac{n^2}{2}
 \end{aligned}$$

for(let end = 0; end < n; end++) {

/ for every end print 0 → end

for(let i = 0; i <= end; i++) {

process.stdout.write(`arr[i]`);

}

console.log();

Improve : [Running stream]

0 1 2 3
[10, 20, 30, 40]

10
10 20
10 20 30
10 20 30 40

let subarr = \varnothing ;

for (let i = 0; i < n; i++) {

subarr = subarr + arr[i] + \varnothing ;

console.log(subarr);

}

* n iterations

* running subarray

subarr = \varnothing

[10, 20, 30, 40] \Rightarrow subarr = \varnothing + 10 = $\{10\}$ ($i=0$)
 $\uparrow \uparrow \uparrow \uparrow$

\Rightarrow subarr = $\{10\} + 20 = \{10, 20\}$ ($i=1$)

\Rightarrow subarr = $\{10, 20\} + 30 = \{10, 20, 30\}$ ($i=2$)

\Rightarrow subarr = $\{10, 20, 30\} + 40 = \{10, 20, 30, 40\}$ ($i=3$)

Q: Generate all subarrays using previous logic

```
let subarr = [];
```

```
for (let i = 0; i < n; i++) {
```

```
    subarr = subarr + arr[i] + " ";
```

```
    console.log(subarr);
```

```
}
```

10

10 20

10 20 30

10 20 30 40

$i=1 \Rightarrow 10 \ 20$
 $=2 \Rightarrow 20 \ 20 \ 30$
 $=3 \Rightarrow 10 \ 20 \ 30 \ 40$

starting with 20

```
let subarr = [];
```

```
for (let i = 0; i < n; i++) {
```

```
    subarr = subarr + arr[i] + " ";
```

```
    console.log(subarr);
```

```
}
```

```
let subarr = [];
```

```
for (let i = 0; i < n; i++) {
```

```
    subarr = subarr + arr[i] + " ";
```

```
    console.log(subarr);
```

```
}
```

$i=2 \Rightarrow 10 + 30$

$i=3 \Rightarrow 10 + 30$

\downarrow
 $10 + 30 + 40$

$= 10 + 30 + 40$

$i=3 \Rightarrow 10 + 40$
 $\rightarrow 10 + 30 + 40$

```
let subarr = [];
```

```
for (let i = 0; i < n; i++) {
```

```
    subarr = subarr + arr[i] + " ";
```

```
    console.log(subarr);
```

```
}
```

```

for (let start = 0; start < n; start++) {
    let subarr = '';
    for (let end = start; end < n; end++) {
        subarr = subarr + arr[i] + ' ';
        console.log(subarr);
    }
}

```

* this will be the template code that we will use for all subarray problems.

* n^2 iterations

0	1	2	3
10	20	30	40

start = 0

\rightarrow end = 0 \rightarrow " 10 \rightarrow "10"

\rightarrow end = 1 \rightarrow "10" 20 \rightarrow "10 20"

\rightarrow end = 2 \rightarrow "10 20" 30 \rightarrow "10 20 30"

\rightarrow end = 3 \rightarrow "10 20 30" 40 \rightarrow "10 20 30 40"

\rightarrow end = 4 (4×4) \times

start = 1

\rightarrow end = 1 \rightarrow " 20 \rightarrow "20"

\rightarrow end = 2 \rightarrow "20" 30 \rightarrow "20 30"

\rightarrow end = 3 \rightarrow "20 30" 40 \rightarrow "20 30 40"

\rightarrow end = 4 /

Find Subarray's with zero sum :

how to get sum of every subarray?

[⁰ ¹ ² ³
10, 20, 30, 40]

```
for (let start = 0; start < n; start++) {  
    let sum = 0;  
    for (let end = start; end < n; end++) {  
        sum = sum + arr[end];  
        console.log(sum);  
    }  
}
```

$$\begin{aligned} \text{start} &= 0 \\ \rightarrow \text{end} &= 3 \\ \rightarrow 0 + 40 &= 40 \end{aligned}$$

start = 0

$$\begin{aligned} \rightarrow \text{end} &= 0 \rightarrow 0 + 10 = 10 \\ \rightarrow \text{end} &= 1 \rightarrow 10 + 20 = 30 \\ \rightarrow \text{end} &= 2 \rightarrow 30 + 30 = 60 \\ \rightarrow \text{end} &= 3 \rightarrow 60 + 40 = 100 \end{aligned}$$

start = 1

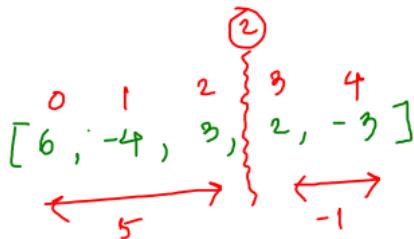
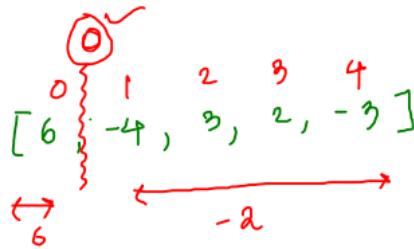
$$\begin{aligned} \rightarrow \text{end} &= 1 \rightarrow 0 + 20 = 20 \\ \rightarrow \text{end} &= 2 \rightarrow 20 + 30 = 50 \\ \rightarrow \text{end} &= 3 \rightarrow 50 + 40 = 90 \end{aligned}$$

start = 2

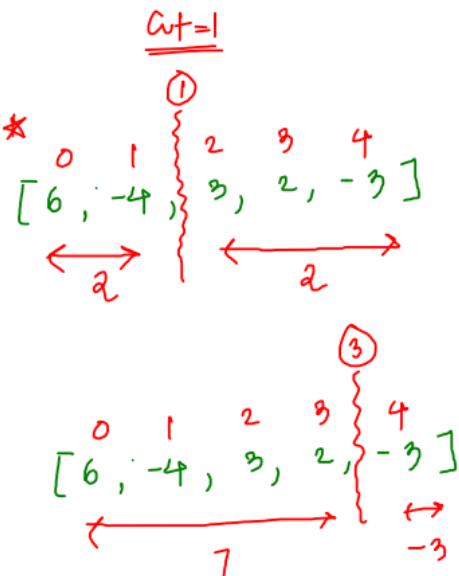
$$\begin{aligned} \rightarrow \text{end} &= 2 \rightarrow 0 + 30 = 30 \\ \rightarrow \text{end} &= 3 \rightarrow 30 + 40 = 70 \end{aligned}$$

op: 10
30
60
100
20
50
90
30
70
40

* Find split point :



* $\text{left} \Rightarrow [0, \text{wt}]$
 $\text{right} \Rightarrow [\text{wt}+1, n-1]$



for(let cut=0; cut < n-1; cut++) {

let lsum = 0;

for(let i=0; i<=cut; i++) {

lsum += arr[i];

}

let rsum = 0;

for(let i=cut+1; i<n; i++) {

rsum += arr[i];

}

if (lsum == rsum) {

return cut;

}

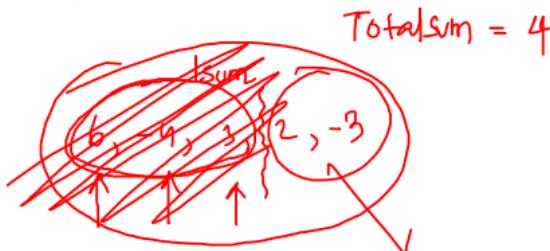
return -1;

* $\text{cut}=0$
 $l+r=n$
 $\text{cut}=1$
 $l+r=n$
 $\Rightarrow n^2$ P. relations

#Improve: [running sum]

$wt = 0$
 $\{ \quad \}$
 $\begin{matrix} 0 & 1 & \{ & 2 & 3 & 4 \\ 6, -4 & \} & 3, 2, -3 \end{matrix}$

$left = [0, wt]$
 $right = [wt+1, n-1]$



$$rsum = total - lsum$$

$\begin{matrix} 6 & \{ & -4 & \{ & 3 & 2 & 3 \\ \uparrow & \uparrow & \} & \} \end{matrix}$

$i=0$

$$lsum = \emptyset \ 6$$

$$rsum = 4 - 6 = -2$$

$i=1$ $lsum = \emptyset \ 2$

$$rsum = 4 - 2 = 2$$

① $i=0$

$$\rightarrow lsum = 0 + 6 = 6$$

$$\rightarrow rsum = 4 - 6 = -2$$

$$\rightarrow 6 - = -2$$

② $i=1$

$$\rightarrow lsum = 6 + (-4) = 2$$

$$\rightarrow rsum = 4 - 2 = 2$$

$$\rightarrow (2 == 2)$$

```

908 function splitArray(N, arr) {
909   let total = 0;
910   for (let i = 0; i < N; i++) {
911     total += arr[i];
912   }
913
914   let lsum = 0;
915   for (let i = 0; i < N - 1; i++) {
916     lsum += arr[i];
917     const rsum = total - lsum;
918     if (lsum == rsum) {
919       return i;
920     }
921   }
922
923   return -1;
924 }
```

$\begin{matrix} 0 & \{ & 1 & 2 & 3 & 4 \\ 6, -4 & \} & 3, 2, -3 \end{matrix}$

$$\Rightarrow total = 4$$

$$\begin{aligned} ① \quad i &= 0 & \rightarrow lsum &= 0 + 6 = 6 \\ && \rightarrow rsum &= 4 - 6 = -2 \\ && \rightarrow 6 - &= -2 \\ ② \quad i &= 1 & \rightarrow lsum &= 6 + (-4) = 2 \\ && \rightarrow rsum &= 4 - 2 = 2 \\ && \rightarrow (2 &= 2) \end{aligned}$$

* Geometric triplets :

Eg: $\begin{bmatrix} 1, 2, 6, 10, 18, 54 \end{bmatrix}$

Ex: $2, \frac{3}{2}, 18$
 $6, \frac{3}{2}, 54$

* generate all triplets
 check whether they
 have same ratio.

$\frac{\text{arr}[j]}{\text{arr}[i]} \times \frac{\text{arr}[k]}{\text{arr}[j]}$

$\rightarrow (\text{arr}[j])^2 = \text{arr}[i] + \text{arr}[k]$

for ($i=0; i < n; i++$) {
 for ($j=i+1; j < n; j++$) {
 for ($k=j+1; k < n; k++$) {
 cl($\text{arr}[i], \text{arr}[j], \text{arr}[k]$);
 }
 }
} } } \rightarrow n^3 \text{ iterations}

$a, b, c, d, e, \dots \dots$

\rightarrow This is in G.P

If $\frac{b}{a} = \frac{c}{b} = \frac{d}{c} = \dots \dots$

(common ratio)

Eg: $1, \frac{2}{3}, \frac{4}{3}, \frac{8}{3}, 16$

$$i=0 \\ \rightarrow j=i+1=0+1=1$$

$$\rightarrow k=j+1=1+1=2 - (0, 1, 2)$$

$$\rightarrow k=3 - (0, 1, 3)$$

$$\rightarrow k=4 - (0, 1, 4)$$

$$\rightarrow k=5 - (0, 1, 5)$$

$$\rightarrow j=2$$

$$\rightarrow k=j+1=2+1=3 - (0, 2, 3)$$

$$\rightarrow k=4 - (0, 2, 4)$$

$$\rightarrow k=5 - (0, 2, 5)$$

Q1: function increment(a) {

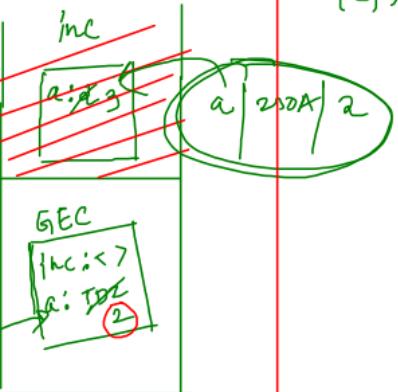
a++;

}

let a = 2;
increment(a);
console.log(a); → 2*

* even the variable names are same, every function will have its own EC
it will have its own variables

name	address	value
a	1B2F	2



Q2: function increment(arr) {

arr[0] ++;

}

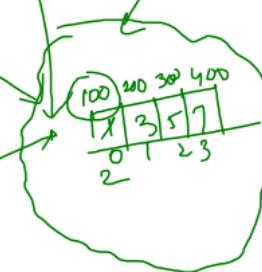
let arr = [1, 3, 5, 7];
increment(arr);
console.log(arr);

[2, 3, 5, 7]

name	address	value
arr	D230F	100

name	address	value
arr	E612A	100

100



pass by reference/address

* changes reflected here unlike the Q1: