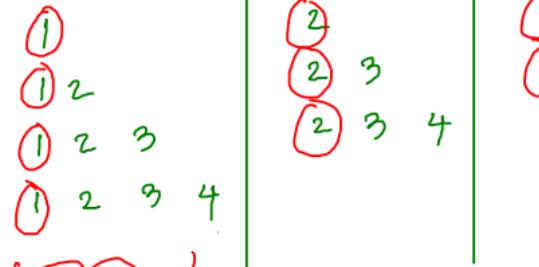


Q: Generate subarray : (any continuous segment of the array)
 (order will be same)

Eg: $[1, 2, 3, 4]$

Op:



η^2 iterations

$$\begin{aligned} 0 &\rightarrow \textcircled{0} \rightarrow 1 \\ 0 &\rightarrow \textcircled{1} \rightarrow 2 \\ 0 &\rightarrow \textcircled{2} \rightarrow 3 \\ 0 &\rightarrow \textcircled{3} \rightarrow 4 \\ \hline &\simeq \eta^2 \end{aligned}$$

$$\begin{aligned} i = \textcircled{0} & \quad j = \textcircled{1} \\ i = \textcircled{1} & \quad j = \textcircled{2} \\ i = \textcircled{2} & \quad j = \textcircled{3} \\ i = \textcircled{3} & \quad j = \textcircled{4} \\ \hline \frac{n(n+1)}{2} & \simeq \eta^2 \end{aligned}$$

how to generate subarrays starting with '10'?

$[10, 20, 30, 40]$

```

for (let i = 0; i < n; i++) {
    for (let j = 0; j <= i; j++) {
        process.stdout.write(` ${arr[i]}`);
    }
    console.log();
}
    
```

ending point

$\Rightarrow \eta^2$

Improving :

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

10

10 20

10 20 30

10 20 30 40

let subarr = " "

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

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

console.log(subarr);

}

\Rightarrow n Iterations

* winning subarray

i = 0 " → "10"

op: 10

i = 1 "10" → "10 20"

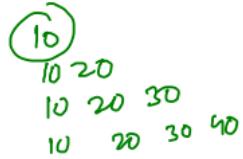
10 20
10 20 30

i = 2 "10 20" → "10 20 30"

10 20 30
10 20 30 40

i = 3 "10 20 30" → "10 20 30 40"

Generate all subarray :
with previous logic :



let subarr = " "

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

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

 console.log (subarr);

}

start = 0

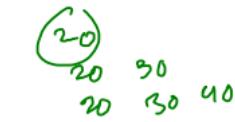
let subarr = " "

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

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

 console.log (subarr);

}



let subarr = " "

(20)
30 40

for (let i = 2; i < n; i++) {

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

 console.log (subarr);

}

start = 2

let subarr = " "

(40)

for (let i = 3; i < n; i++) {

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

 console.log (subarr);

}

start = 3

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

```
    let subarr = " "
```

```
    for ( let end = start; end < n; end++ ) {
```

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

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

```
}
```

```
}
```

op: 10

10 20

10 20 30

10 20 30 40

20

20 30

20 30 40

(0 , $\boxed{1 \ 2}$, 3
10 , $\boxed{20 \ 30}$, 40)

start = 0 (arr subarr with 10)

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

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

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

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

end = 4 X end < n (4 < 4)

start = ① (subarr start with 20)

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

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

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

end = 4 X

```

1131 function zeroSubarray(arr) {
1132   const n = arr.length;
1133   let isFound = false;
1134
1135   for (let start = 0; start < n; start++) {
1136     let subArrSum = 0;
1137     for (let end = start; end < n; end++) {
1138       subArrSum = subArrSum + arr[end];
1139       // you will have the sum of [start, end]
1140       if (subArrSum == 0) {
1141         isFound = true;
1142         console.log(`Subarray found from Index ${start} to ${end}`);
1143       }
1144     }
1145
1146     if (isFound == false) {
1147       console.log(-1);
1148     }
1149   }
1150 }
```

$\text{start} = 0 \quad \text{sum} = 0$
 $\text{end} = 0 \rightarrow \text{sum} = 0 + 3 = 3$
 $\text{end} = 1 \rightarrow \text{sum} = 3 + 4 = 7$
 $\text{end} = 2 \rightarrow \text{sum} = 7 + (-7) = 0 \quad (0, 2)$
 $\text{end} = 3 \rightarrow \text{sum} = 0 + 3 = 3$
 $\text{end} = 4 \rightarrow \text{sum} = 3 + 1 = 4$
 $\text{end} = 5 \rightarrow \text{sum} = 4 + 3 = 7$

$\text{start} = 1, \text{sum} = 0$
 $\text{end} = 1, \text{sum} = 0 + 4 = 4$
 $\text{end} = 2, \text{sum} = 4 + (-7) = -3$
 $\text{end} = 3, \text{sum} = -3 + 3 = 0 \quad (1, 3)$
 $\text{end} = 4, \text{sum} = 0 + 1 = 1$
 $\text{end} = 5, \text{sum} = 1 + 3 = 4$

Q: Find split point :

eg: $[6 \underset{6}{\overset{0}{\leftarrow}} -4 \underset{-4}{\overset{1}{\leftarrow}} 3 \underset{3}{\overset{2}{\leftarrow}} 2 \underset{2}{\overset{3}{\leftarrow}} -9 \underset{-9}{\overset{4}{\leftarrow}}]$



*
* $[6 \underset{2}{\overset{0}{\leftarrow}} -4 \underset{-4}{\overset{1}{\leftarrow}} 3 \underset{3}{\overset{2}{\leftarrow}} 2 \underset{2}{\overset{4}{\leftarrow}} -9]$



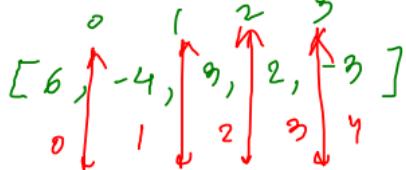
Op: 1

$$\begin{bmatrix} 6 & -4 & 3 & 2 & -9 \\ \swarrow & \searrow & \downarrow & \downarrow & \downarrow \\ 5 & & -1 & & \end{bmatrix}$$

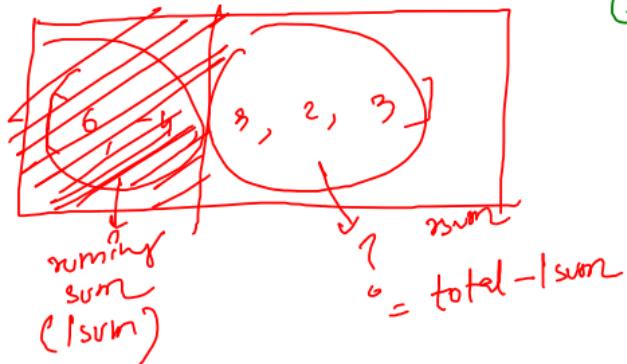
$$\begin{bmatrix} 6 & -4 & 3 & 2 & -9 \\ \swarrow & \searrow & \downarrow & \downarrow & \downarrow \\ 7 & & -3 & & \end{bmatrix}$$

```
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;
```

Improve : (running sum)



$$\text{sum} \neq 2 \neq 8 \neq 1$$



$$\rightarrow n + n - 1$$

$$\rightarrow 2n - 1 \approx n \text{ iterations}$$

$$\begin{aligned} \textcircled{1} \quad & \text{total} \\ &= 4 \end{aligned}$$

$$\textcircled{2} \quad \text{cut} = 0$$

$$\begin{aligned} \text{lsum} &= 6 \\ \text{rsum} &= 4 - 6 \\ &= -2 \end{aligned}$$

$$\begin{aligned} \textcircled{3} \quad \text{cut} &= \textcircled{1} \\ \text{lsum} &\approx 6 + (-4) \\ &= \textcircled{2} \\ \text{rsum} &\approx 4 - \textcircled{2} \\ &= \textcircled{2} \\ &\text{(return)} \end{aligned}$$

$$\cancel{\textcircled{4}} \quad \text{cut} = 2$$

```

1206 function findSplit(n, arr) {
1207   let total = 0;
1208   for (let i = 0; i < n; i++) {
1209     total += arr[i];
1210   }
1211
1212   let lsum = 0;
1213   for (let cut = 0; cut < n - 1; cut++) {
1214     lsum += arr[cut];
1215     const rsum = total - lsum;
1216     if (lsum == rsum) {
1217       return cut;
1218     }
1219   }
1220
1221   return -1;
1222 }
```

$$\cancel{\textcircled{5}} \quad \text{cut} = 2$$

$$\begin{aligned} \text{lsum} &= 2 + 3 \\ &= 5 \end{aligned}$$

$$\begin{aligned} \text{rsum} &\approx 4 - 5 \\ &= -1 \end{aligned}$$

$$\begin{aligned} \textcircled{6} \quad \text{lsum} &= 5 + 2 \\ &= 7 \end{aligned}$$

$$\text{rsum} \approx 4 - 7 = -3$$

$$\cancel{\textcircled{7}} \quad \text{cut} = 4$$

$$\begin{aligned} 4 &< n - 1 \\ 4 &< 5 - 1 \\ 4 &\times 4 \times \cancel{X} \end{aligned}$$

Q: Geometric triplets:

q: [1, 2, 6, 10, 18, 54]

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

→ are in G.P

→ common ration

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

$\begin{matrix} 2 & 6 & 10 \\ \swarrow & \searrow & \\ 3 & 3 \end{matrix}$

* how to generate
triples
(any 3 numbers)

$\begin{matrix} 1 & 2 & 4 & 8 & 16 \\ \swarrow & \searrow & \swarrow & \searrow & \swarrow \\ 2 & 2 & 2 & 2 \end{matrix}$

$\begin{matrix} 6 & 18 & 54 \\ \swarrow & \searrow & \\ 3 & 3 \end{matrix}$

```
for(let i=0; i<n; i++) {  
    for(let j=i+1; j<n; j++) {  
        for(let k=j+1; k<n; k++) {
```

}

}