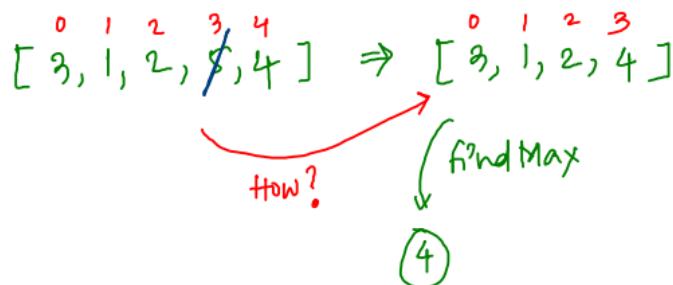


* ^{2nd} Largest Element :

eg: [3, 1, 2, 5, 4]

Op: 4

- remove 1st largest
- find the max among the rem. elements



then how do I remove pt largest?

= we will not actually remove it but we will just ignore it,

[3, 1, 2, 5, 4]

- ① firstMax = 5
- ② Again run the find max But this time ignore the ele = firstMax

[3, 1, 2, ~~5~~, 4]

maxEle = -
5 = first
contain (Ignore)

* Reverse an Array :

Eg: [10, 20, 30, 40, 50]

Op: [50, 40, 30, 20, 10]

function reverse (arr) {

/*
 code
*/

*/

}

const arr = [1, 2, 3, 4, 5];

reverse (arr);

console.log (arr) // [5, 4, 3, 2, 1]

#1:

Const rev-arr = [] ;

for (let i = n-1; i >= 0; i--) {

rev-arr.push (arr[i]);

}

Ex [50] [50, 40] [50, 40, 30] [50, 40, 30, 20]
[50, 40, 30, 20, 10]

* as the changes need to be done in arr, copy the elements

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

[1, 2, 3, 4, 5]

5 4 3 2 1

arr

arr[i] = rev-arr[i];

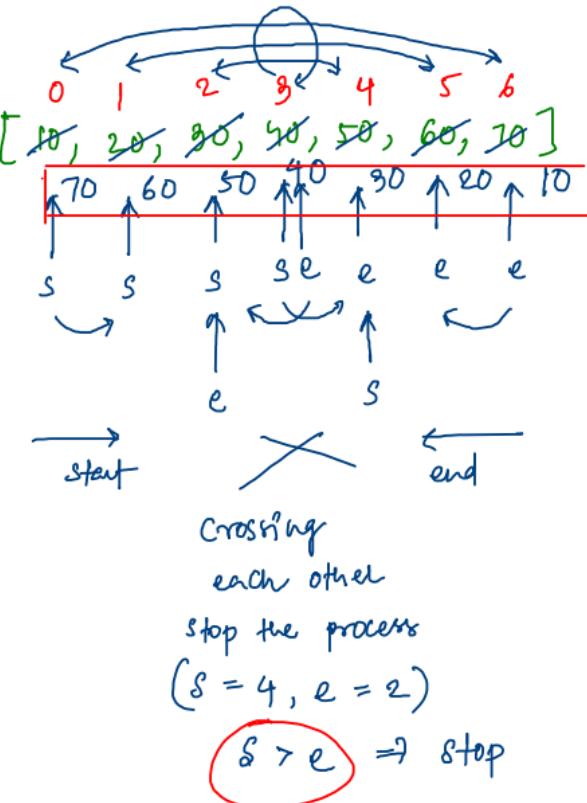
}

#2 : Two - pointer Approach

In-place
solution

arr =

↓
make changes within
the given array
Instead of taking
an extra array.



let start = 0 ;
let end = n-1 ;

while (start <= end) {
 swap (a [s] , a [e]) ;
 s++ ;
 e-- ;
}

* Sum of Array except self :

Eq: [4, 3, 2, 10]

$$op: [15, 16, 17, 9]$$

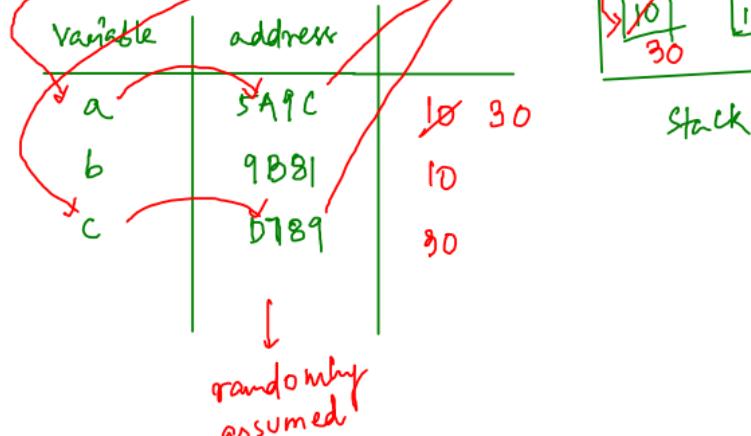
The diagram shows the four numbers 15, 16, 17, and 9. Arrows point from each number to the corresponding operations: 15 and 16 both point to $+2+10$, 17 points to $+3+10$, and 9 points to $+3+2$.

$$1. \text{ totalSum} = 4+3+2+10$$

$$2^{\circ} \quad [4, 3, 2, 10] \quad \text{total - auxil} \quad \begin{array}{l} (4) + (3) + (2) + (10) = 19 \\ \underbrace{(4)}_{\text{sum}} + (3) + (2) + (10) = 19 \end{array}$$

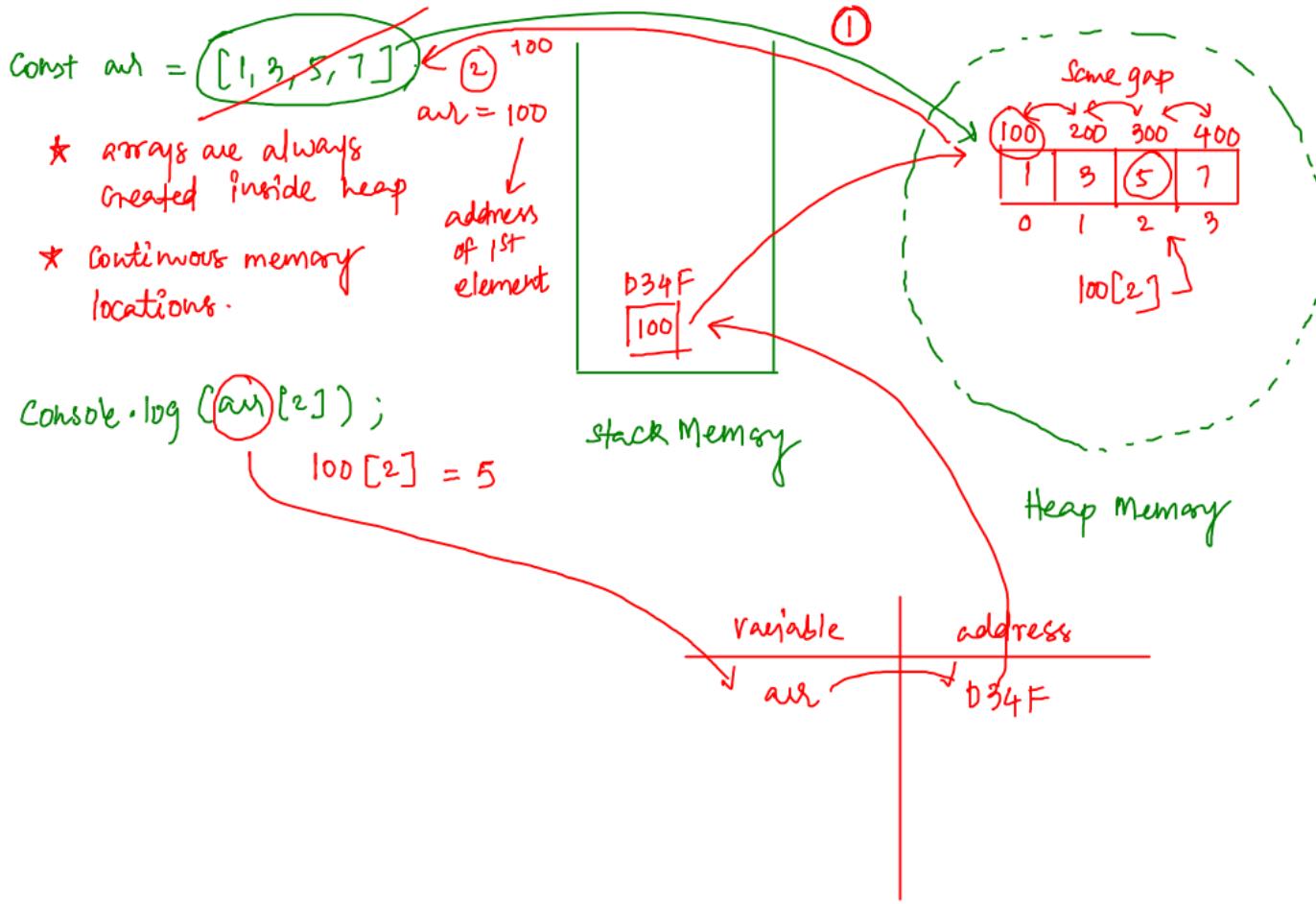
* How arrays work Internally ?

```
let a = 10;  
console.log(a);  
let b = a; → b=10  
let c = 30;  
a = c; → a=30
```



* address in memory & in hexadecimal format (base-16)

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

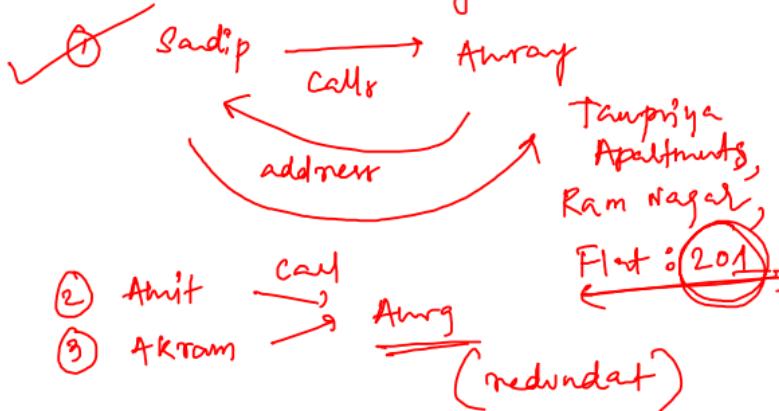


30	301	302	303
201	202	203	
101	102	103	

* 3 friends have purchased flats, (201, 202, 203) (Sudip, Amit, Akram)

* Let say you are a common friend to these 3 friends (Anurag)

* house warming ceremony,

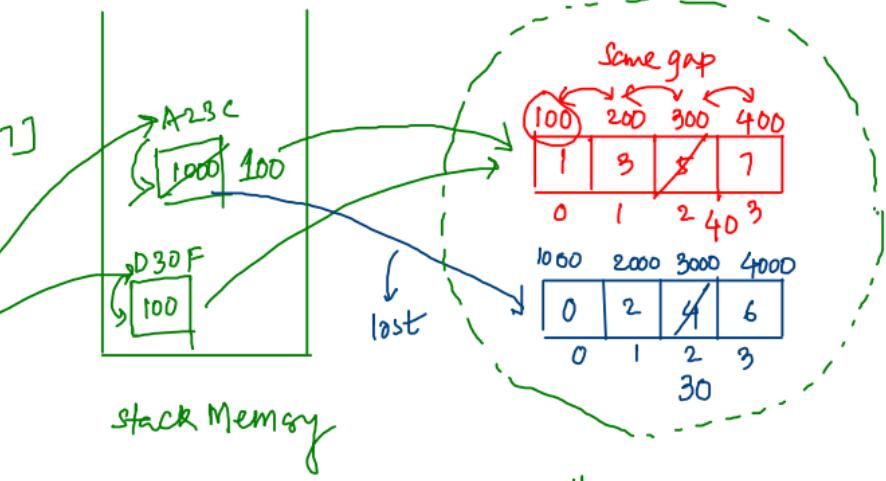


⇒ With only Sudip's address Anurag will know Amit, Akram's address as well

201 → 202 → 203
 wkt they are best friends.

Example :

```
100  
791 let arr = [1, 3, 5, 7];  
792 console.log(arr);  
793  
794 let brr = [10, 20, 40, 60];  
795 brr[2] = 30;  
796  
797 brr = arr;  
798 arr[2] = 40;  
799 console.log(brr);
```



$$\star \text{ brr}[2] = 30 \\ \downarrow \\ 1000[2] = 30$$

$$\star \text{ brr} = \text{arr} \\ \text{brr} = 100$$

$$\star \text{ arr}[2] = 40 \\ \text{arr}[2] = 40$$

* even though arr,
brr are different variables
they are looking at same array.

* `console.log(brr);` $\Rightarrow [1, 3, 40, 7]$

* pass - by - value

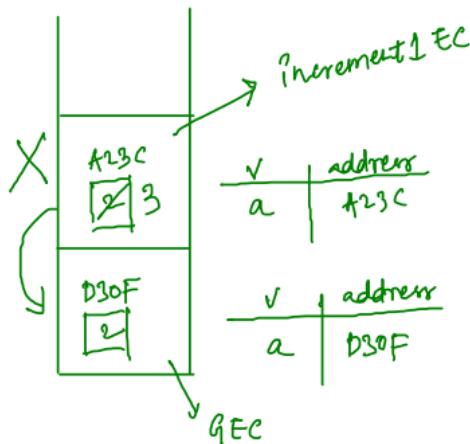
```

801 function increment1(a) {
802   a++;
803 }
804 let a = 2;
805 increment1(a);
806 console.log(a);
807
  
```

$a = 2$

not Same

2



* Every EC / function will have its own memory space.

* Pass - by - reference

* $brr[0]++$
 \downarrow
 $100[0]++$

```

809 function increment2(brr) {
810   brr[0]++;
811 }
812 let arr = [1, 3, 5, 7];
813 increment2(arr);
814 console.log(arr);
815
  
```

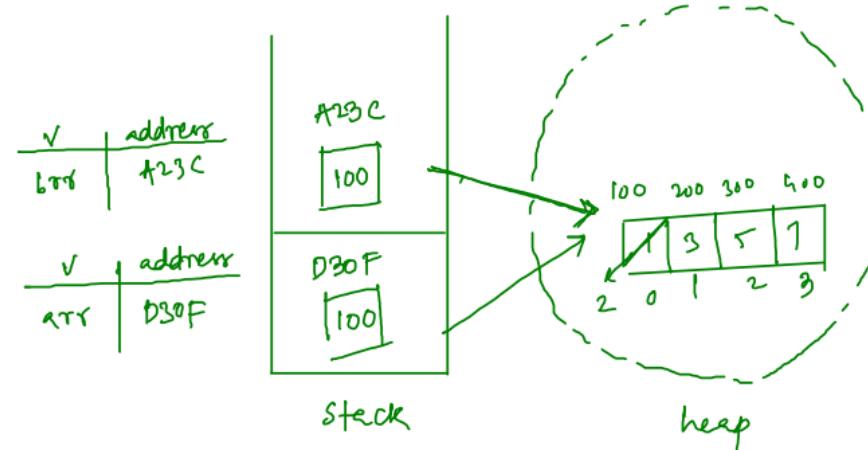
Same

$brr = 100$

100

address / reference

$[2, 3, 5, 7]$



* changes will be reflected after function call.

* Array Problem 6 :

Ex: $\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 2, 4, 1, 6, 7 \end{bmatrix}$

\Rightarrow Min dist b/w any 2 positive even numbers

$$\begin{array}{ll} \textcircled{1} & (2, 4) \\ \Rightarrow & (0, 1) \\ \Rightarrow & \text{abs}(0-1) = \textcircled{1} \end{array}$$

$$\textcircled{2} (2, 6) \Rightarrow (0, 3) \Rightarrow \alpha_5(0-3) = 3$$

* min dist
↳ only possible
for nearer
elements

* How to generate any two pairs

```
let mindiff = Infinity;
```

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

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

* If $(\text{and } [i] \cdot r_2 = 0 \text{ & } \text{and } [j] \cdot r_2 = 0)$ {

and $i > 0$ const dist = Math.abs(i - j);
 & and $j > 0$ minDiff = Math.min(minDiff, dist)

} } $\simeq n^2$ iterations
 } }
 → optimize to n .