

Array
Street how how

→
arr = [1, 2, 3, 4, 5]

index 0 , 1 , 2 , 3 , 4
Floor no

{ arr[0] → 1

arr[3] → 4

arr.length - 1
i →

Const arr = [5, 3, 6, 8, 1, 7, 9, 2, 4]

for (let index in arr)

con. log (index , arr[index])

data access

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

[] ?

index → data where the data

is

for in

for of. → data
in → index

~~Index starts from 0~~

Index starts from 0

Enden ends in len - 1

→ []

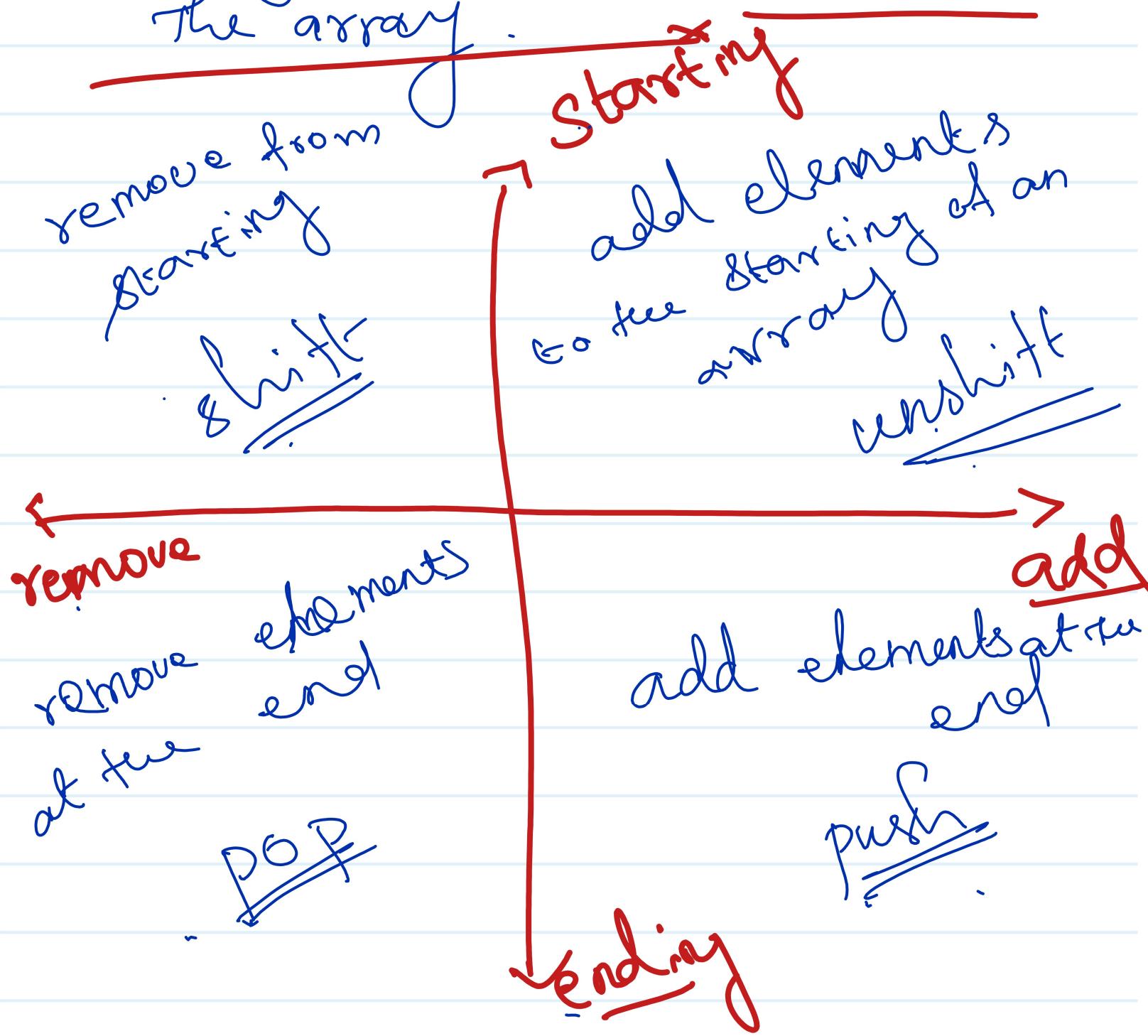
000 → 0 99

0000 → 0 999

negative? no

→ ↗

if Index of method returns
-r means, the value you're
looking for is not in
the array.



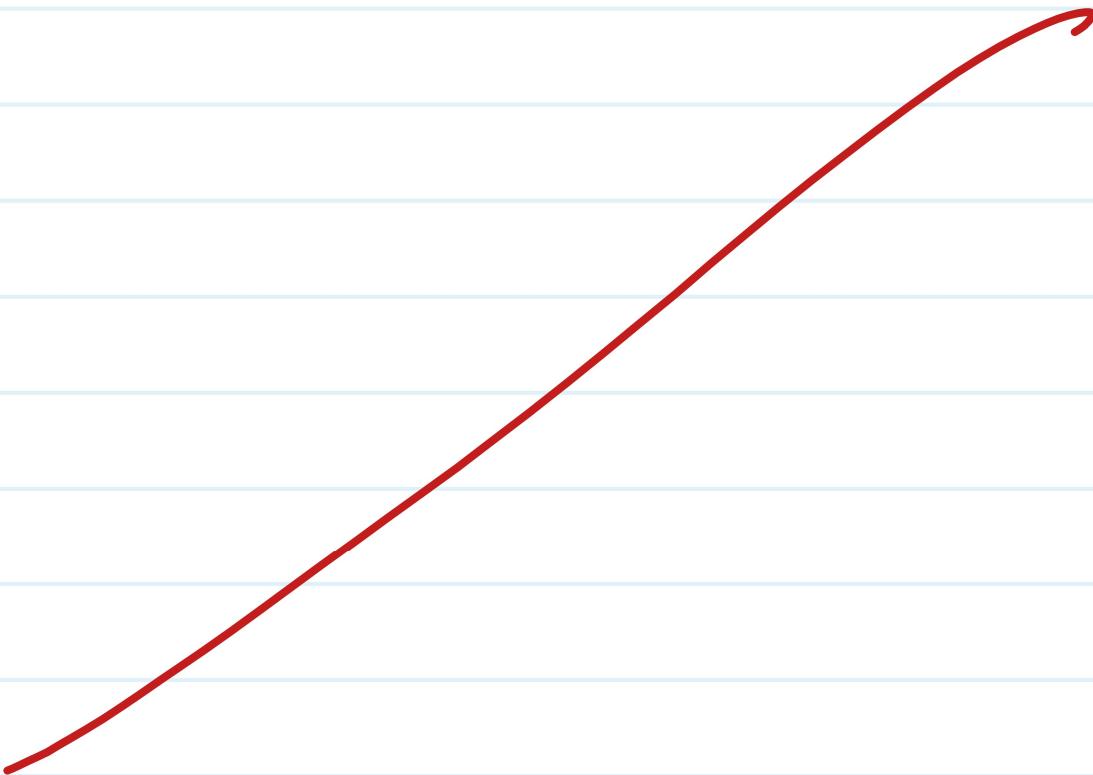
0 1 2 3 4 5 6 7 8

const arr = [6, 7, 8, 9, 10, 11, 12, 13, 14]

console.log(arr) [6, 7, 8, 9, 10, 11, 12, 13, 14]

const copiedPortion = arr.slice(2, 6)

console.log(copiedPortion) [8, 9, 10, 11]



MRF

Map ->

Creates a new array
with results of every
function call for each
element

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

```
const res = arr.map(val => {
    return val * 2;
```

3

```
console.log(arr) "[1, 2, 3, 4, 5]"
console.log(res) "[2, 4, 6, 8, 10]"
```

map -> arr.length := res.length

Filter ->

- Creates a new array with ^{few or more} the values from the original array based on the values returned from function

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

const odd = arr.filter(val => {

if (val % 2 == 1)

return true;

else

return false;

})

for 1, 3, 5 alone

we will get true

Console.log(carry)[1, 2, 3, 4, 5]
Console.log(odd)[1, 3, 5]

filter ↗

0 <= res.length <= arr.length



Reduce:

res.length == 1