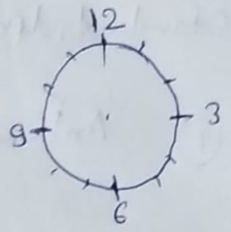
Coding Practice → CodewithNishant ✓

↓

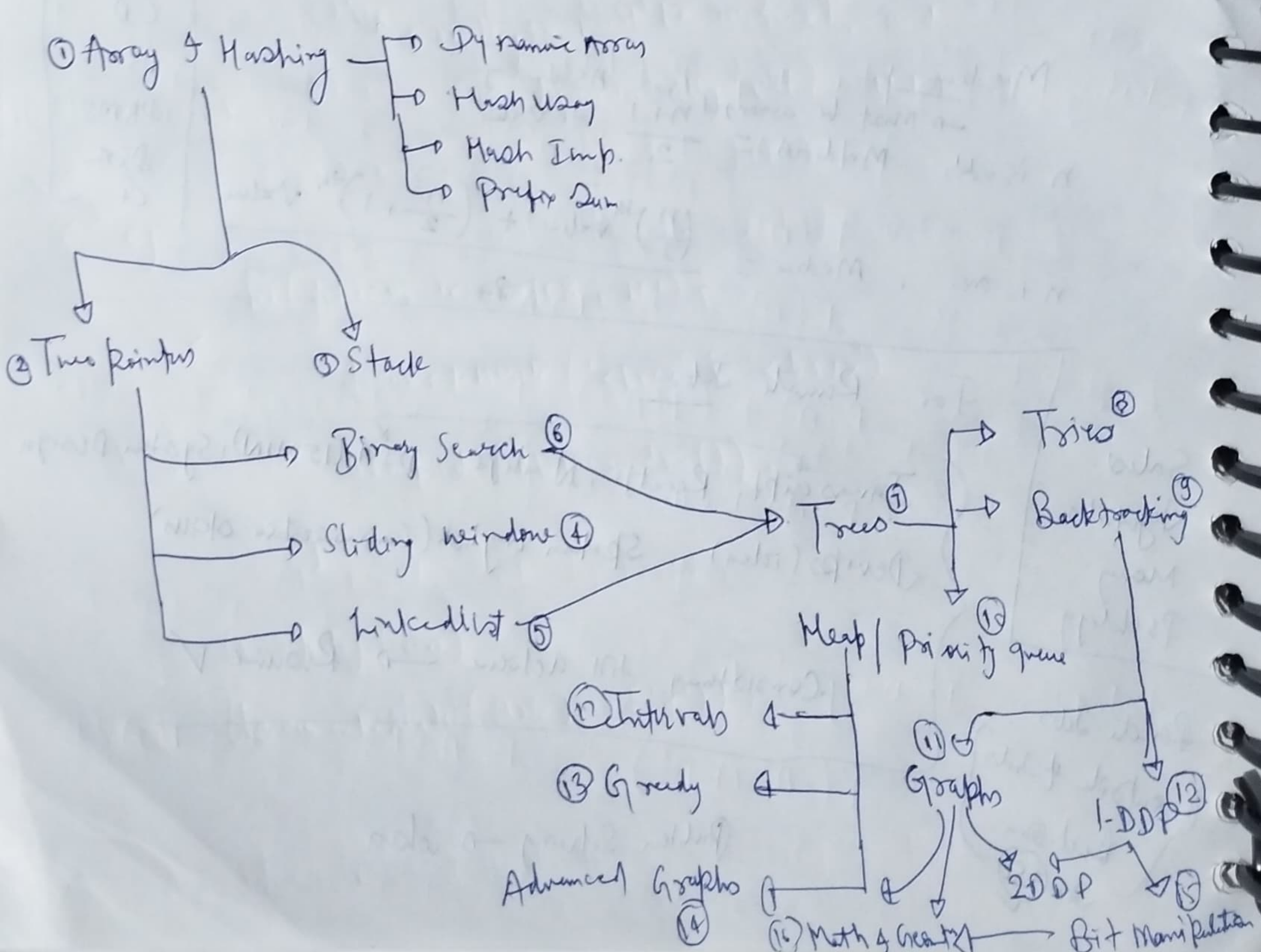
Revision → Basic → Method, Tricks, Trif, How solve in DSA
→ Javascript | React | Node.js | System basic for express for interview

JS ✓
React
Nodejs
DSA → 15 Min.
System Design.
Spoken English (Communication Skills)
Devops
Resume → Apply
Twitter



Roadmap

Follow JS → Make Notes from Namaste JS (core concept) | Follow with 10-12-23



Array & Hashing

Prefix Sum

$$\begin{array}{rcccccl} & 0 & 1 & 2 & 3 & 4 & = i \\ \text{arr} = [& 0, & 1, & 6, & 3, & -1 &] \\ \text{prefixArr} = [& 0, & 1, & 7, & 10, & 9 &] \end{array}$$

Range = A[0] ⟶ B[4]

A[0] → 0   Sum of arr = 9

A[0,1] ⟶ A[2,4]

Summation ⟹ A[0,1] + A[2,4] = B[4] ⟺

1 + 8 = 9

B[4] − A[2,4] = A[0,1]

or

or, B[4] − A[0,1] = A[2,4]

or, A[2,4] = B[4] − A[0,1]

So, $A[i,j] = A_j - A[0,i-1]$

$$\boxed{A[i,j] = A_j - A[0,i-1]}$$

$A[2,4] = A_4 - A[0,1]$

Pseudo Code

```
let arr = [0,1,6,3,-1]
Find prefixSum Arr = ?
let prefixSumArr[0] = arr[0]
for(let i = 1; i < arr.length; i++){
    prefixSumArr[i] = prefixSumArr[i-1]
        + arr[i]
}
Console.log(prefixSumArr)
```

Dyanmic Array in JS

```
let arr = new Array()

let arr = new Array('Elon', 'Musk', 'Steve') → ['Elon', 'Musk', 'Ste

let arr = new Array(108)
```

↳ It creates empty array which have undefined values, its length will be 108

```
let arr = []  ← this is also array creation,

arr.push('Ram')
arr.push('Shyam')
arr.push('Mohan')
```

↳ arr length will be increased, and its dyanamically value added to end.

eg.

```
for( let i=0; i<=108 ; i++) {
    document.write('<br/>' i '<br/>')
    arr.push(Math.random());
}

console.log(arr)
document.write(arr.join('>'));
```

Hash Map in JS

Hashmap in Javascript, we are mostly used object. its here similar as hash-map.

Hash map $\longleftrightarrow$ { }

let hashmap = { }

$$\begin{tikzcd} & \{\} \arrow[dl, dash] \arrow[dr, dash] & \\ \text{Key} & & \text{Value} \end{tikzcd}$$ Pairs

hashmap['tomato'] = 3

hashmap ['Rose'] = 1

```
{ 'tomatto' : 3 ,
  'Potato'  : 9 ,
  'Rose'    : 1 ,
  'Pen'     : 3
}
```

Key	Value
tomatto	3
Potato	9
Rose	1
Pen	3

e.g.

let arr = ['tomatto', 'Potato', 'Rose', 'Pen']

Make Hash map,

```
let hash = {}
  for( let el of arr) {
      if(!hash[el]) {
          hash[el] = 1
      }
      else {
          hash[el] = hash [el] +1
      }
```

Here, If undefined, element will be added, its initial value will be 1

If not undefined, its value will be increment by 1

for(let element in arr)

$\downarrow$

it just iterate on key.

Object.keys( )

object.values( )

object.entries( )

Other question in Hash

eg. let hash = {
    'Civil' : ['Ram', 'Mohan']
    'Mechanical' : ['Subham']
    'CS' : ['Rohit']
    'ECE' : ['Elon', 'Stone', 'Sunil']

}

How, we can create ?

Problem:- The have Json from backend, where we first store in key value pairs and after, show on screen.

```
Const json = [{ name: 'Ram', branch: 'Civil'},
              { name: 'Subham', branch: 'Mechanical'},
              { name: 'Rohit', branch: 'CS'},
              { name: 'Elon', branch: 'ECE'},
              { name: 'Stone', branch: 'ECE'},
              { name: 'Sunil', branch: 'ECE'}
             ]
```

From Backend,

Show on Screen

I meant, sort by branch name.

$$\begin{tikzcd} \end{tikzcd}$$

Multiple check

/^[A-Za-z]+$/ } - Regex

^ → Start; [A-Za-z] → upper case lower case; $ → end

Sol:- Const hashStore = { }

idea → First iterate the array, in every iteration check branch name is different, so yes, so we can add [], and if same then Push.

```
Cost hash = { }

for( let i = 0; i < arr json.length ; i++) {

    ~~if( Object. Keys~~
    ~~if (!hash [i.branch]) {~~

        ~~hash [i.branch] = [ i.name]~~

        }

    else {
        ~~hash [i.branch] = [ ... i.name]~~

        }

}
```

```
if(! hash [arr[i].branch]) {

    hash [arr[i].branch =

            [arr[i].name];

    }
else {

    hash [arr[i].branch].push

        ( arr[i].name)

    }
```

✓

P-2 :- let s = 'abcdefa'; find number of a element ; Ans a = 2

```
Const hash = { }

    for( let i = 0; i < s.length ; i++) {

            if (! hash [s[i]]) {

                    hash [s[i]] = 1

            }

            else {
                    hash [s[i]] = hash [s[i]] + 1

            }

    }
```

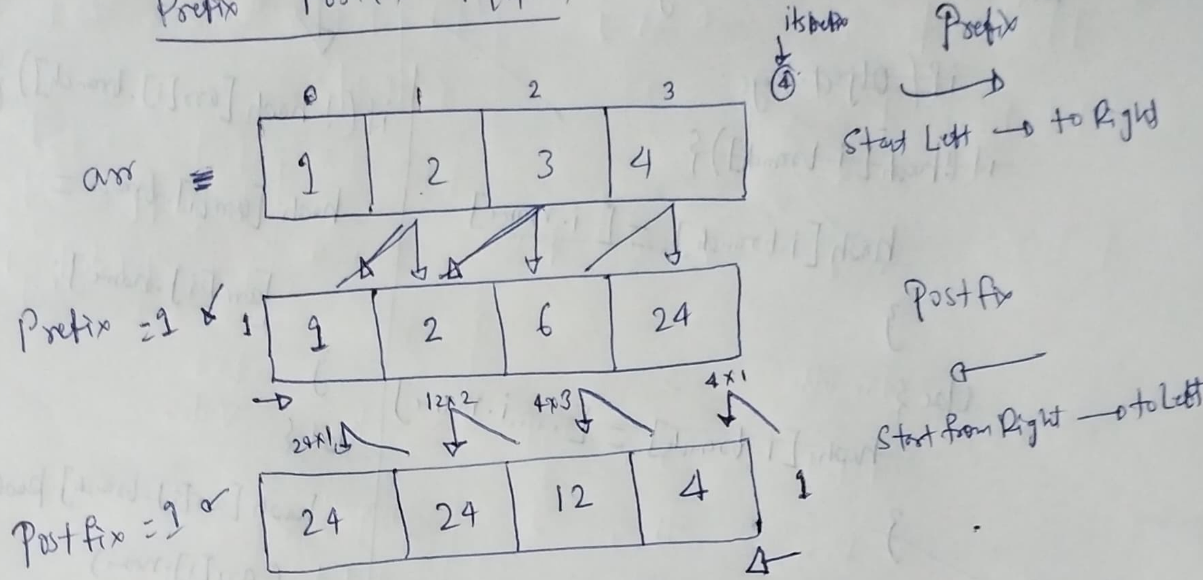
Console.log (Object.Values (hash)) ⟶ find Value in array form

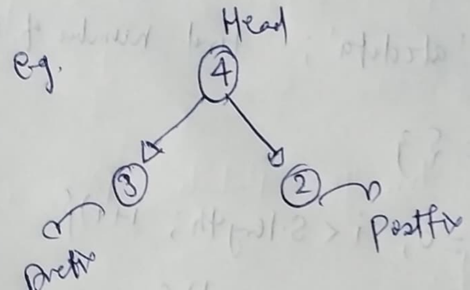
Console.log (Math.max (...Object.value (hash))); ⟶ Here, max value

Here, we itarate over hash object, but here clarify, find a, so just write
Console.log (hash ['a']) // 2

Contains Duplicate | Two Sum | Valid Anagram

Prefix Postfix Method





→ If you, finding of 2 index, Prefix, So for that, just multiply previous prefix index i.e 1. and then multiply by Current index i.e. 2 index of actual array.

So,

Prefix of array of 2 index, or

$arr[2] = Prefix[2-1] \times arr[2]$

⇒ Prefix of arr[2] = 2 × 3

$$\boxed{Prefix[2] = 6}$$

$$\boxed{Prefix[n] = Prefix[n-1] \times arr[n]} \quad n > 0$$

For Postfix,

→ it is same as prefix, but we will start from right side of array.

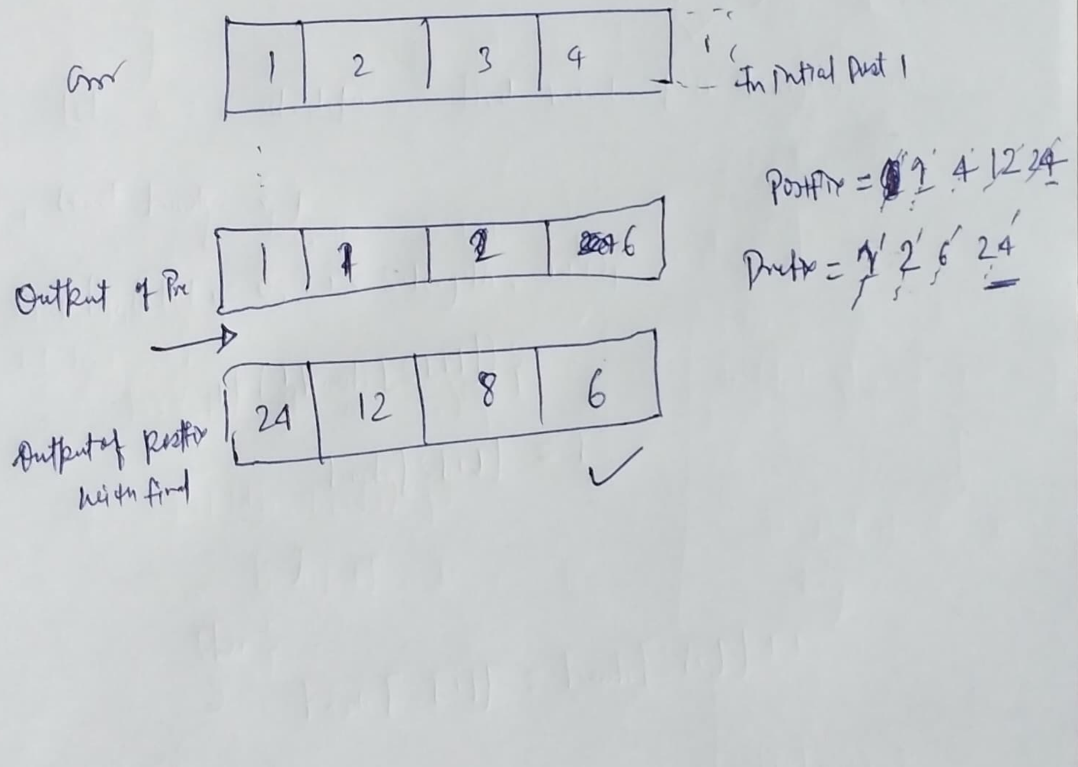
Suppose, Postfix of index 3.

So, we assume in starting, Postfix = 1 or prefix = 1. reason is, we have no any element.

i.e. postfix [3] = postfix (3+1) × arr[3]

$$\boxed{\text{postfix}[n] = \text{postfix}[n+1] \times \text{arr}[n]}$$

Final Output, By Actual Code,


arr
1
2
3
4
Initial Post 1
Postfix = 1 4 12 24
Prefix = 1 2 6 24
Output of Pre
1
1
2
6
Output of Postfix with final
24
12
8
6
✓