Agenda

(*) Spread, rest and default parameter [...arr]

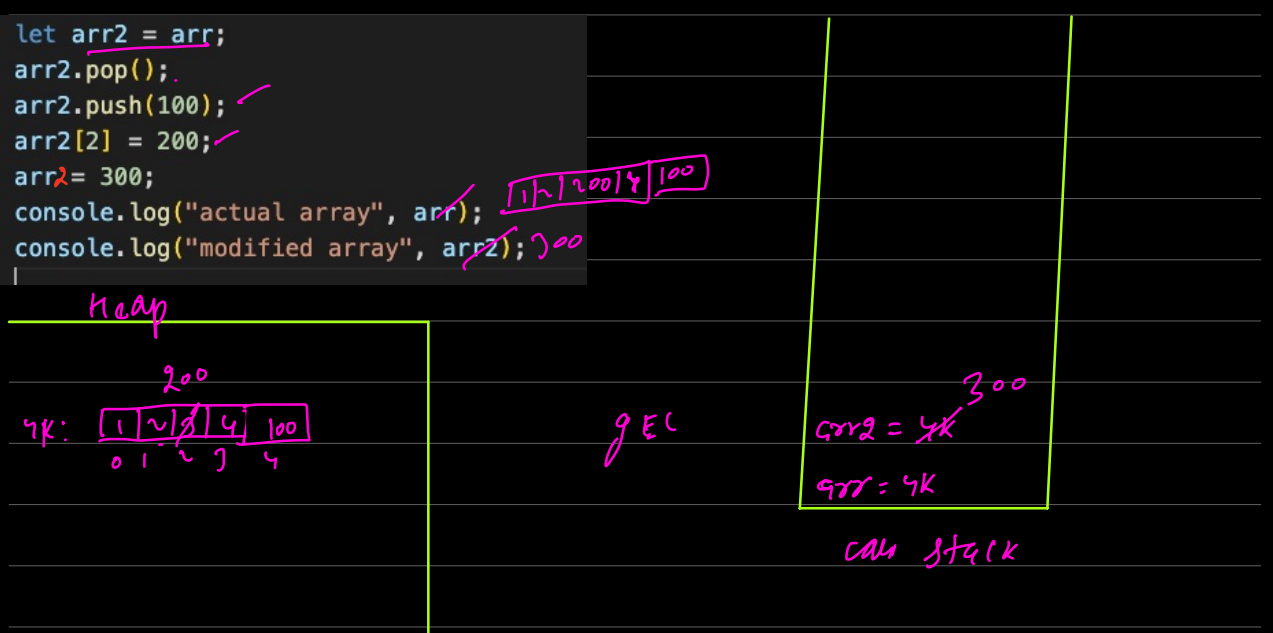
(*) polyfills of bind, call apply.

(*) call by ref and call by value

(*) deep copy and shallow copy

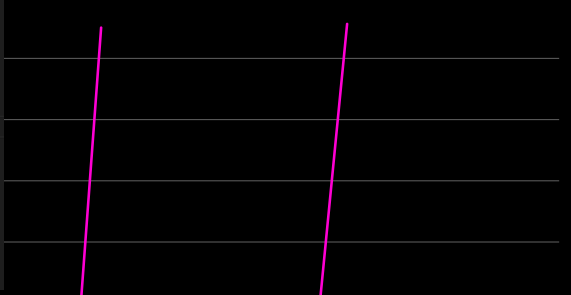
(*) flatten an array.

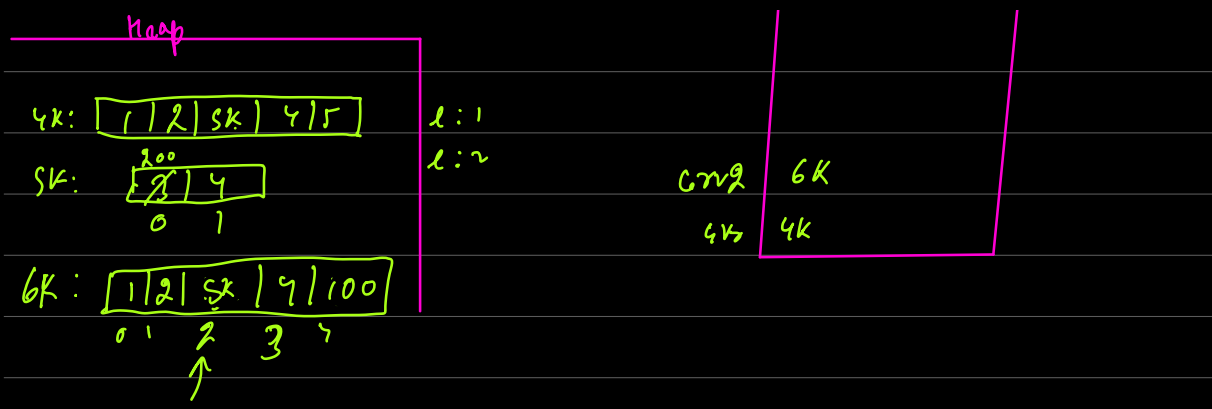
```
let arr2 = arr;
arr2.pop();
arr2.push(100);
arr2[2] = 200;
arr2= 300;
console.log("actual array", arr);
console.log("modified array", arr2);
```



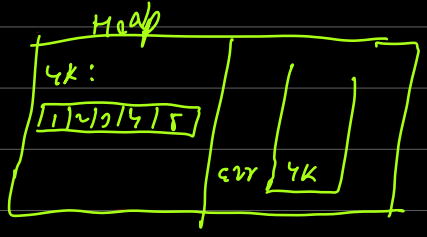

```
let arr = [1, 2, [3, 4], 4, 5];
// // spread copies value  from one array another array for the first level
let arr2 = [...arr];
arr2.pop();
arr2.push(100);

// let arr = [1, 2, [3, 4], 4, 5];
// [value,value,ref,value,value]
// arr2[2] = 100;
arr2[2][0] = 200;

console.log("orginal arr", arr, "updated array", arr2);
```

= SK(0) = 200



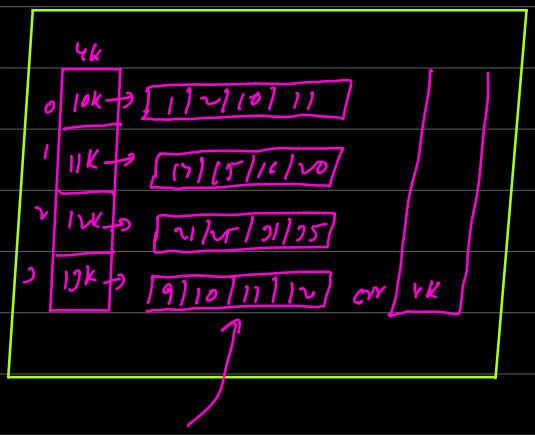


(*) arr =



arr =

1	2	10	11
14	15	16	20
21	25	31	35
9	10	11	12

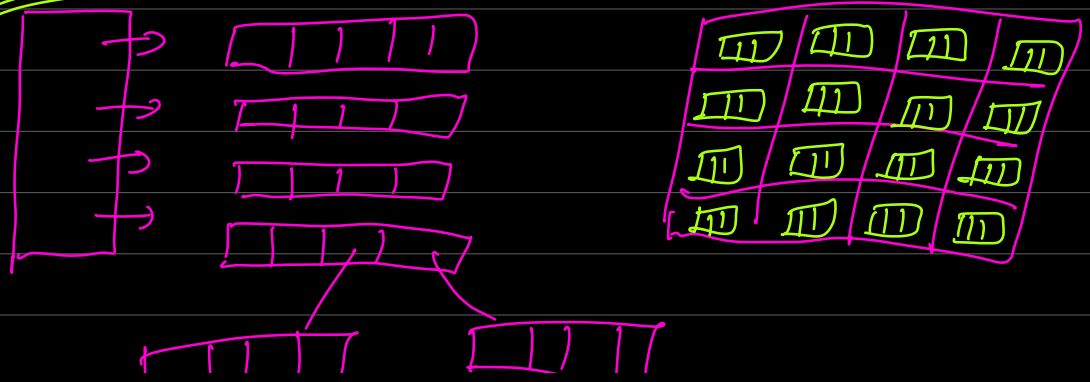


arr[3].length

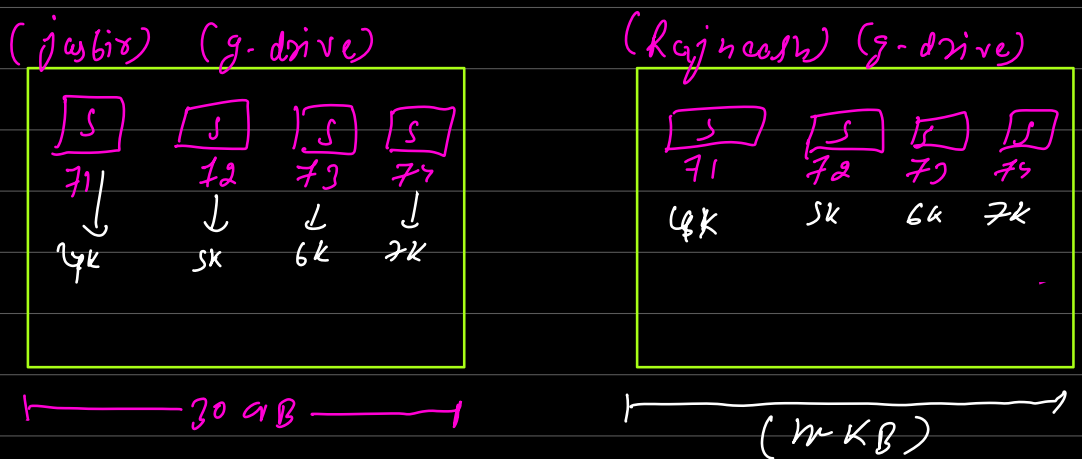
↓

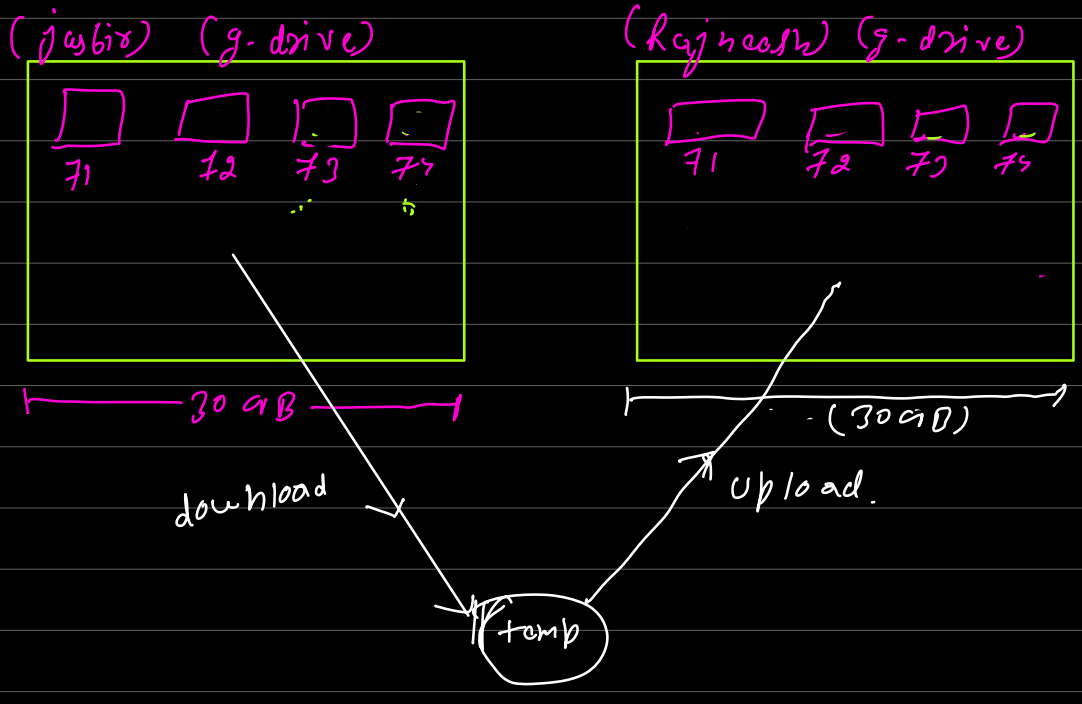
13K.length ≡ 9|10|11|12 .length.

2D-Array



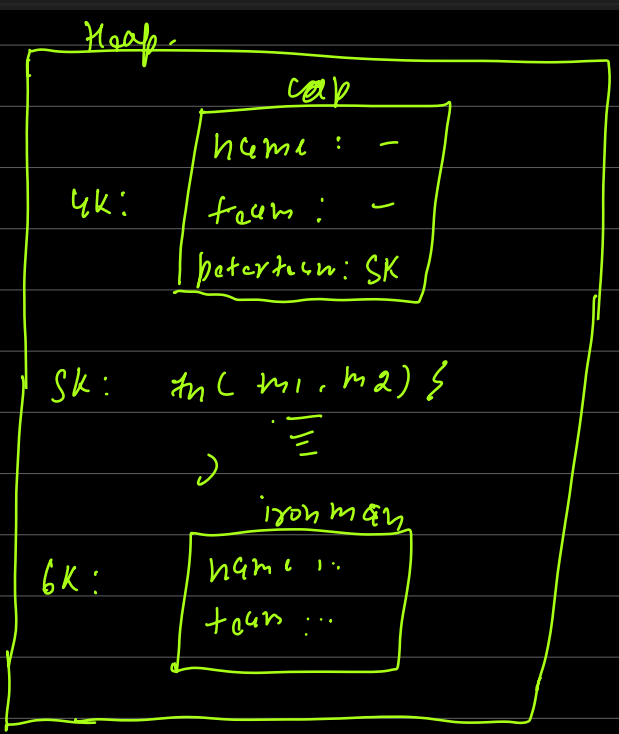
Shallow copy



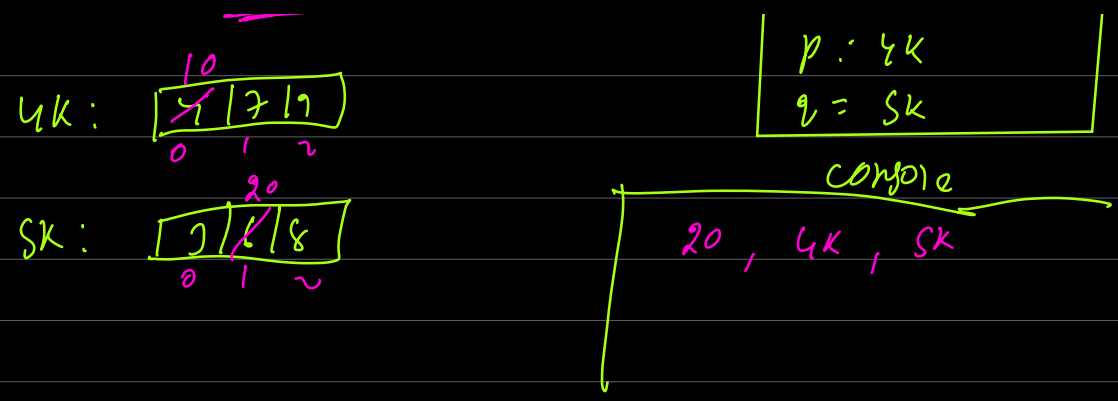


(Deep copy / Hard copy) : which you copy each bit of memory segment.

```
let cap = {
    name: "Steve",
    team: "Cap",

    petersTeam: function (mem1, mem2) {
        console.log(`Hey ${this.name} I am your neighborhood's
        spiderman and i belong to ${this.team}'s team with members  ${mem1} and ${mem2}`);
    }
}
let ironMan = {
    name: "Tony",
    team: "Iron Man",
}
```




```
function modifier(a, b) {
    console.log("13", a, b)
    a[0] = 10;
    b[1] = 20;
    console.log("16", a, b)
}
let p = [4, 7, 9]
let q = [3, 6, 8]

console.log("20", p, q);
modifier(p, q)
console.log("23", p, q);
```


4K:
10
7
17
0
1
2
5K:
20
8
0
1
2
p : 4K
q = 5K
console
20, 4K, 5K