→ [[9 – 12] [8 – 9] [7 – 12] [3 – 5] [13 – 15]
[6 – 10] [7 – 10]]

↳ if we have non overlapping ranges then meeting rooms Can be **reused**.

[[9 - 12] [8 - 9] [7-12] [3 - 5] [13-15]

[6 - 10] [7 - 10]]

→ To allocate a room, we need to be sure that no other room is free.

Start → [9, 8, 7, 3, 13, 6, 7]

end → [12, 9, 12, 5, 15, 10, 10]

→ Sort asc

Start → [9, 8, 7, 3, 13, 6, 7]

end → [12, 9, 12, 5, 15, 10, 10]

i

Merge 2 sorted arrays

Start → [3, 6, 7, 7, 8, 9, 13]

end → [5, 9, 10, 10, 12, 12, 15]

j

→ Current Meeting Rooms In Use → ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~3~~ ~~4~~ ~~2~~
~~2~~ ~~1~~ ~~0~~ 1

<

→ Result → ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

```
while ( i< start.length and j< end.length) {
        if ( end[j] < = start[i]) {
                current--;
                j++;
        } else {
                current++;
                result = max (result ,current);
                i++
        }
}
```

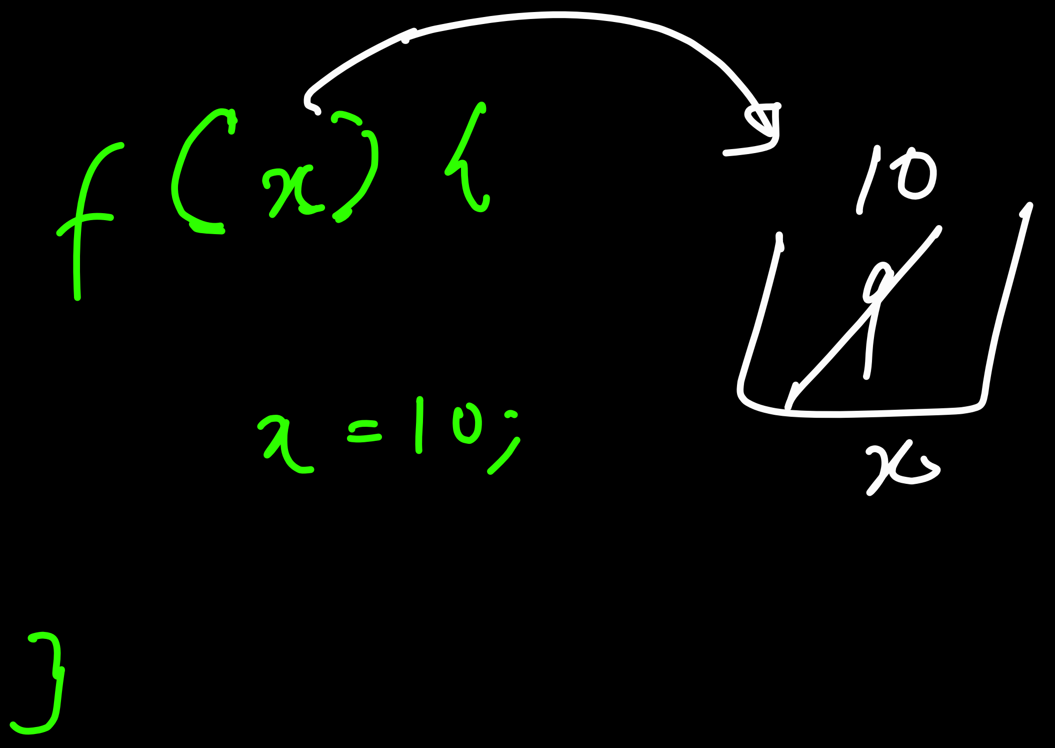
2j

intervals= [[1, 3] [4, 6] [9, 10] [3, 5]]

x =

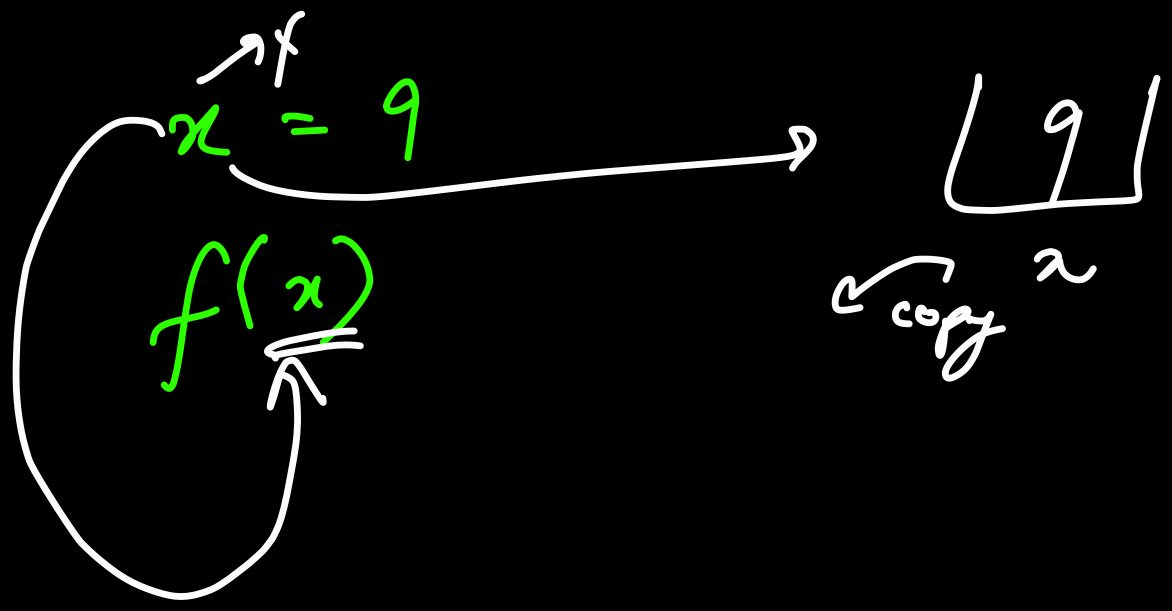
```
i = interval.length -1
j = intervals.length -2
while (j >= 0 and interval[j][0] >
          interval[i][0] )<

}
```



pass by value

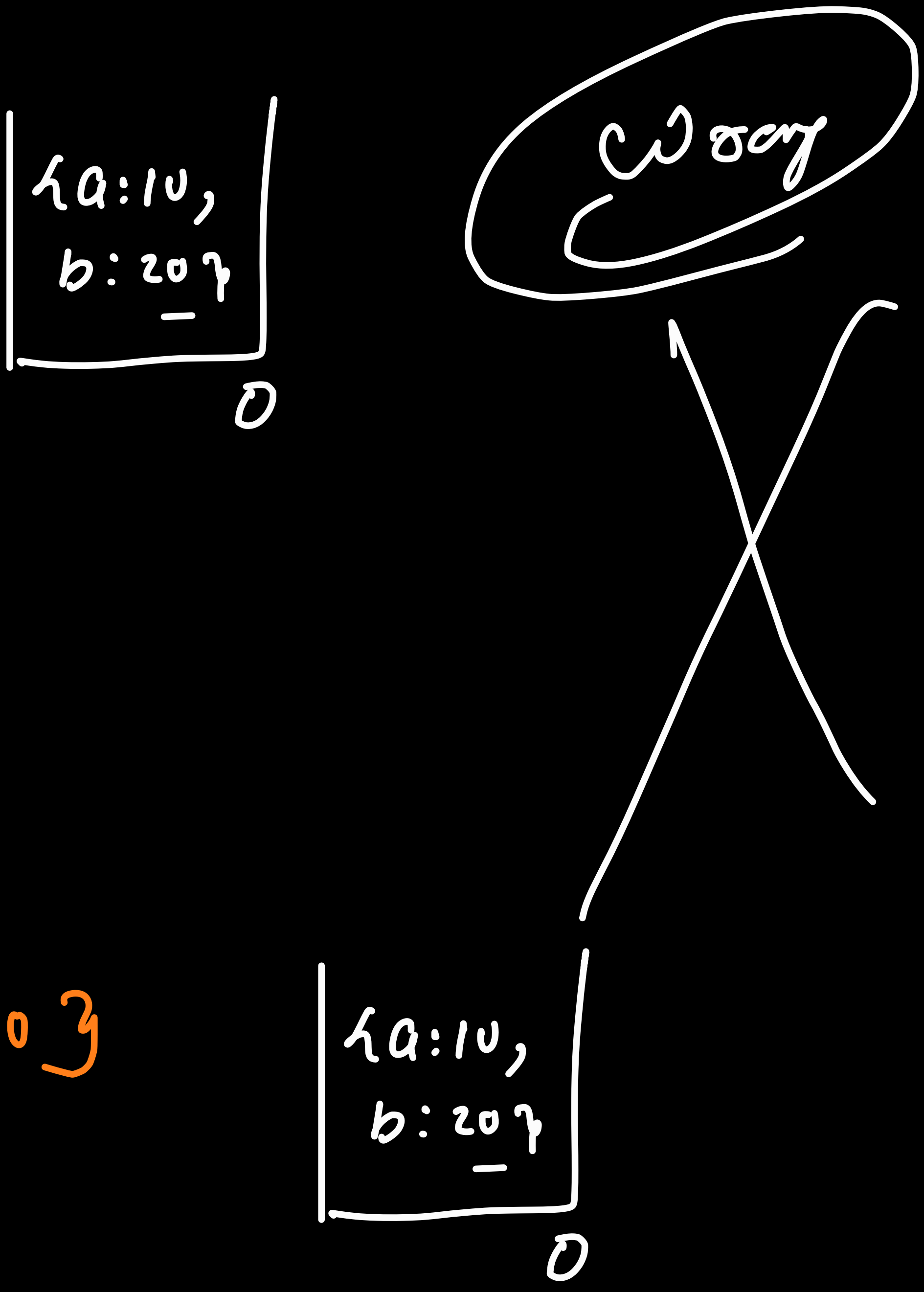
pass by copy



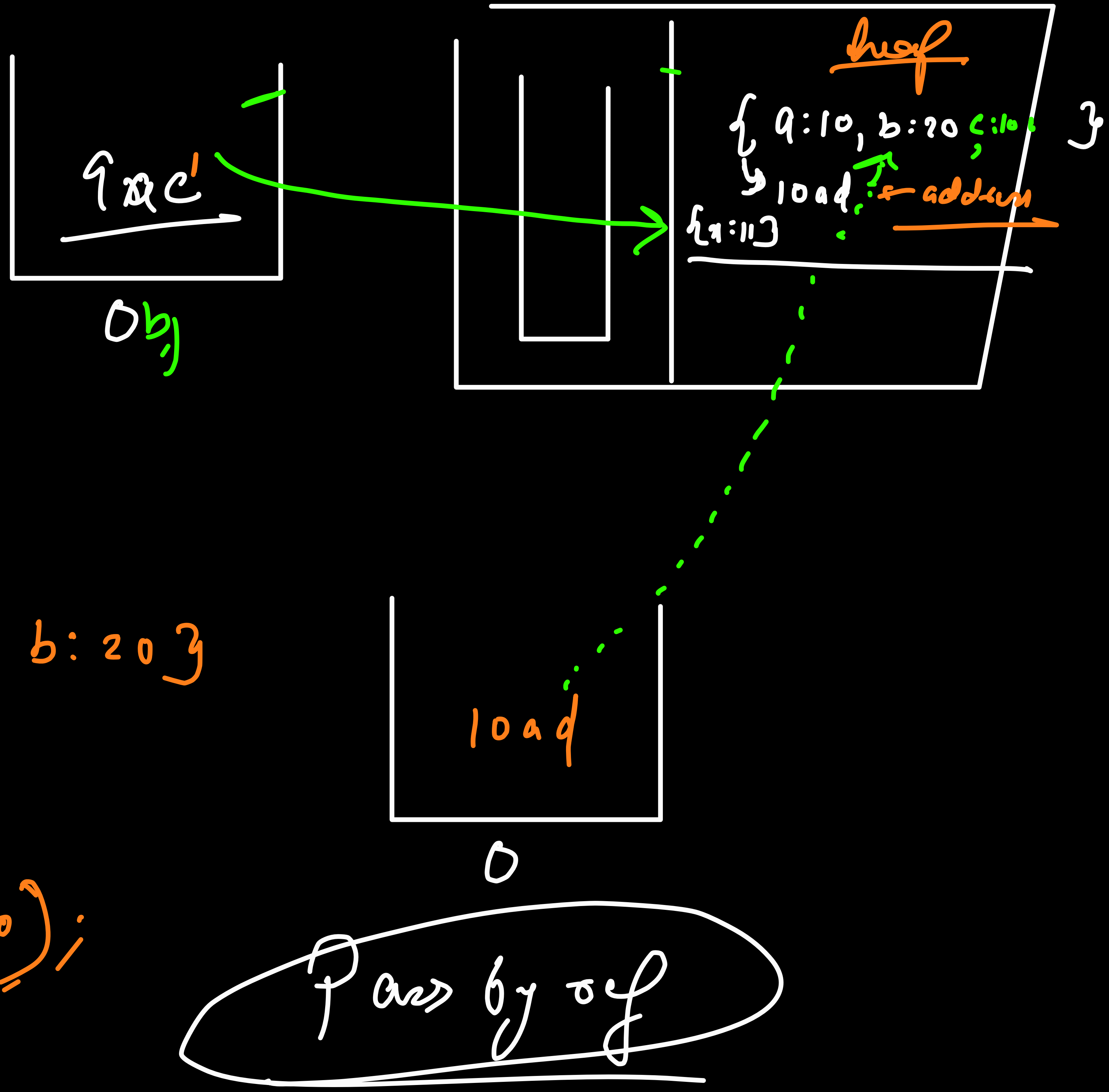
```
f(obj) {


}


o = {a:10, b:20}
f(o)
console.log(o);
```



```
f(obj){
    obj.c=100;
    Obj={x:11}
}
```



```
o = {a:10, b:20}
f(o)
console.log(o);
```

Pass by ref