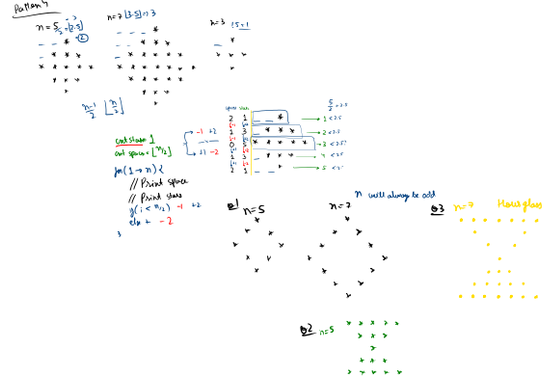
Iterative algo
Arrays
2d Arrays
Strings

Recursion
Trees
LinkedList

Stacks
Queues




```
function pattern3(n: number) {
    for (let i = n; i >= 1; i--) {
        let s: string = "";

        let countStars = i;
        let countSpaces = n - i;

        //I have to add countSpaces
        while(countSpaces > 0){
            s+=" ";
            countSpaces--;
        }
        //I have to add stars
        while(countStars > 0){
            s+="*";
            countStars--;
        }

        console.log(s);
    }
}
```

```
* * *
 * *
```

Iterative algo

Q1 n = 5

```
1 *
2 * *
3 * * *
4 * * * *
5 * * * * *
```

n = 3

```
1 *
2 * *
3 * * *
```

3

n = lines

No. of stars = line no.

```
for (let i = 1; i <= n; i++) {
    s = ""
    for (let j = 1; j <= i; j++) s += '*';
    console.log(s)
}
```

3

```
function pattern1(n: number) {
    for (let i = 4; i <= 3; i++) {
        let s: string = "";
        for (let j = 1; j <= i; j++) s += "*";
        console.log(s);
    }
}
```

```
*
* *
* * *
```

10 < 5 → False

10 > 10 → False

10 >= 10 → True

10 > 5 → True

Pattern 2

n = 5

```
5 * * * * *
4 * * * *
3 * * *
2 * *
1 *
```

n = 3

```
* * *
* *
*
```

Pattern 3

n = 5

```
i = 5 * * * * *
4 _ * * * *
3 _ _ * * *
2 _ _ _ * *
1 _ _ _ _ *
```

n = 3

```
* * *
_ * *
_ _ *
```

stars = i 3

spaces = n - i 5 - 3 = 2