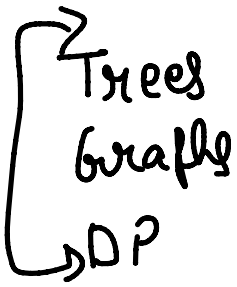
Recursion

20 August 2023 13:02

- Function calls itself.
- When sol. of a problem depends on sol. of smaller instances of same problem.



(i) Base Case

(ii) Recursive Case

Factorial

20 August 2023 13:18

$5! = 120$ $(5 \times 4 \times 3 \times 2 \times 1)$

$(5 \times 4!)$

for(i=n; i>=1; i--)
{
} = loop

$n! = n * (n-1)!$

Base case

if(n == 0)
return 1;

$5! = 5 \times 4!$

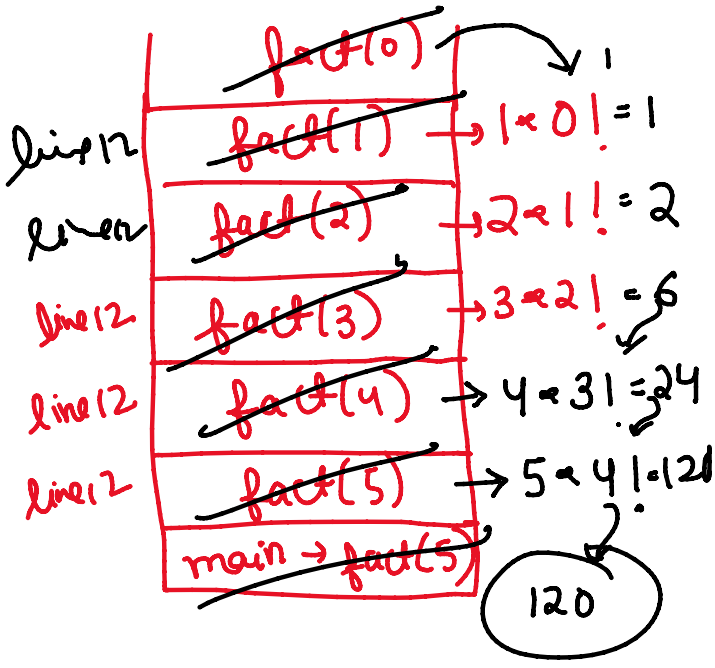
$4 \times 3!$

$3 \times 2!$

$2 \times 1!$

$1 \times 0!$

$n = 5$



Base case:

n == 0

① Figure out the base case.

② Assume that subproblems will be solved automatically

① [illegible]

② Assume that subproblems will be solved automat

③ Using subproblems write and for current problem

Time - O(N)

Space - O(N)

O(1)

Fibonacci series

20 August 2023 13:36

Find n^{th} fibonacci number.

→ Sum of previous two numbers.

n = 6

ans = 8

a b a b c

0 1 1 2 3 5 8 13 21 .....

0th 1st 2nd 3rd 4th 5th

n = 5

c = a + b

Base case

```
if(n==0||n==1)
    return n;
```

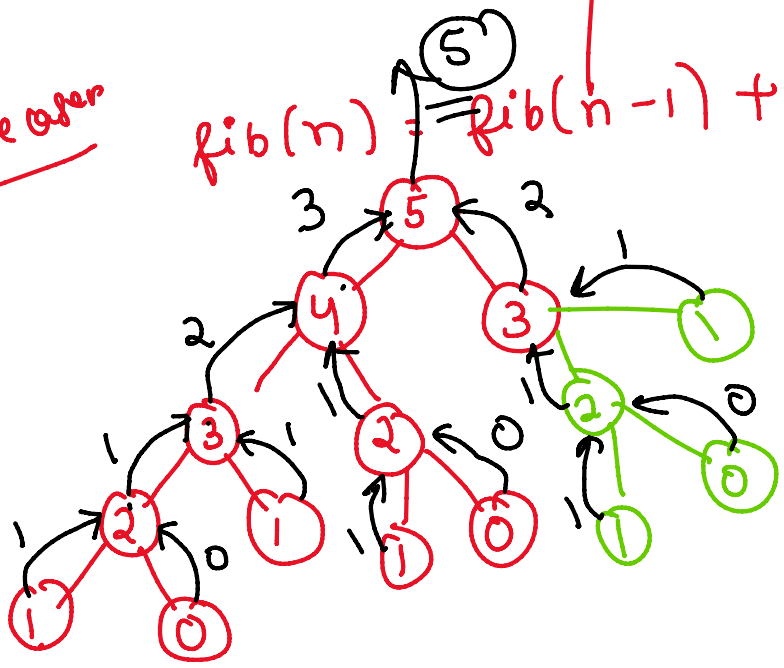
```
if(n<=1)
    return n;
```

ans = 5

away → O(N) → space

Recursive case

fib(n) = fib(n-1) + fib(n-2)



1

2

4

8

16

Time : $O(2^N)$

Space : $O(N)$

Left to right direction.

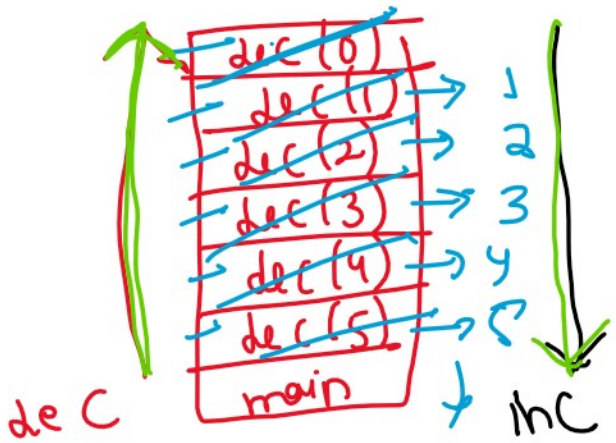
Inc dec order

20 August 2023 13:50

n = 5

dec → 5, 4, 3, 2, 1

5, 4, 3, 2, 1

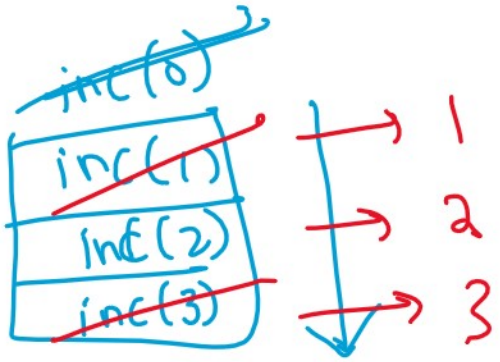


incr → 1, 2, 3, 4, 5

Time → O(N)
Space → O(N)

Base case
if (n == 0) return;

```
void dec (int n)
{
    print(n); //dec
    dec(n-1);
    print( )//inc.
}
```



1, 2, 3

Power

20 August 2023 14:00

a^b x^n

a, b

a = 5, b = 3

ans = $5^3 = 125$

$5^0 = 1$

$10^0 = 1$

Base case:

```
if(b==0)
return 1;
```

||

```
if(b==1)
return a;
```

recursive case

power(a, b) = a * power(a, b-1);

Time = O(N)

Space = O(N)

$5^3 = 5 * 5^2$ ↷ 25

$5^2 = 5 * 5^1$ ↷ 5

$5^1 = 5 * 5^0 \Rightarrow 1$.

125

odd/even

power (2,5)

$5^{16} \rightarrow 5^{15} \rightarrow 5^{14} \rightarrow 5^{13} \cdots\cdots 5^{0}$ O(N)

$5^{16} \rightarrow 5^{8} \rightarrow 5^{4} \rightarrow 5^{2} \rightarrow 5^{1} \rightarrow 5^{0}$ O(logN)

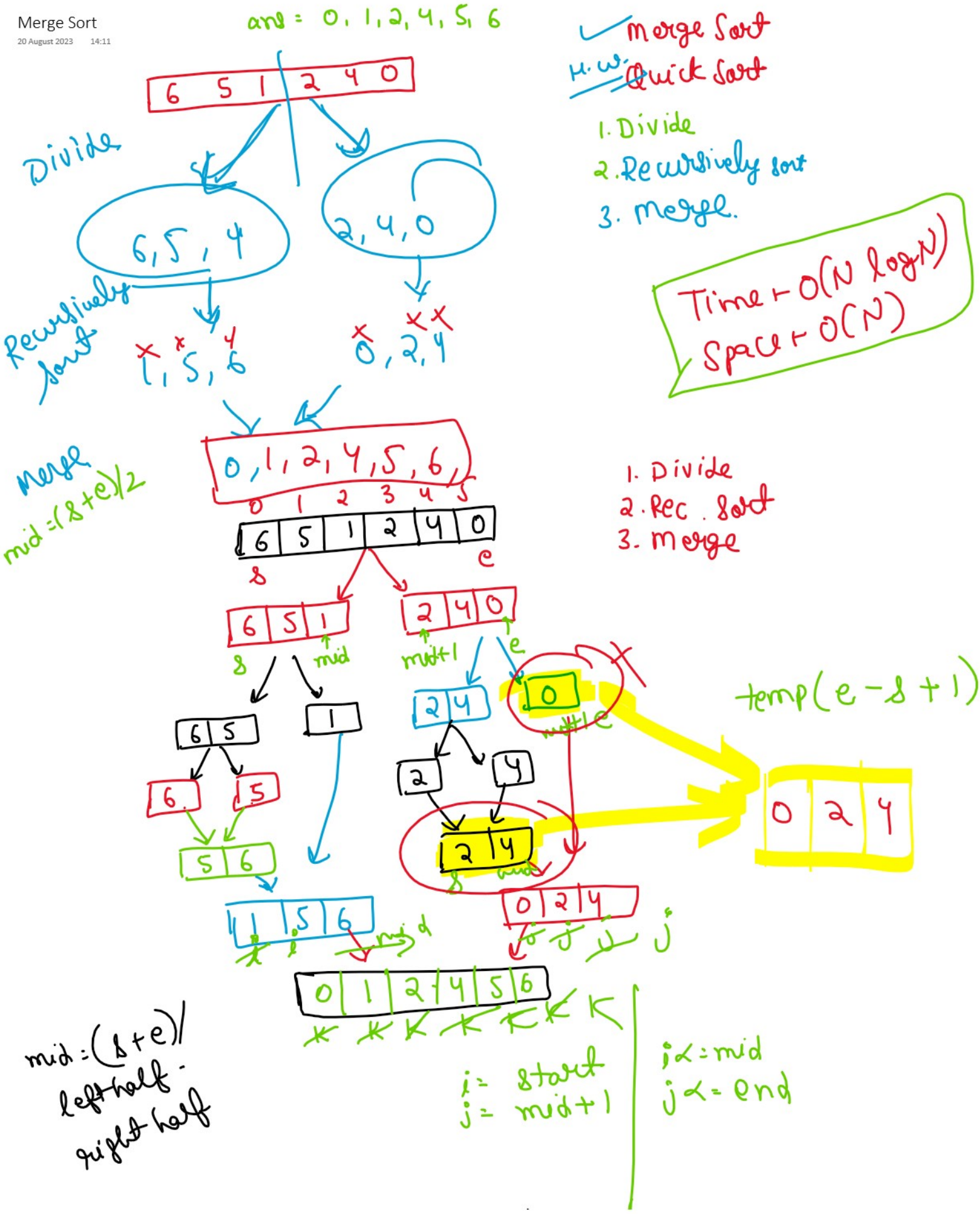
even b%2 == 0
return power(a, b/2) * power(a, b/2);

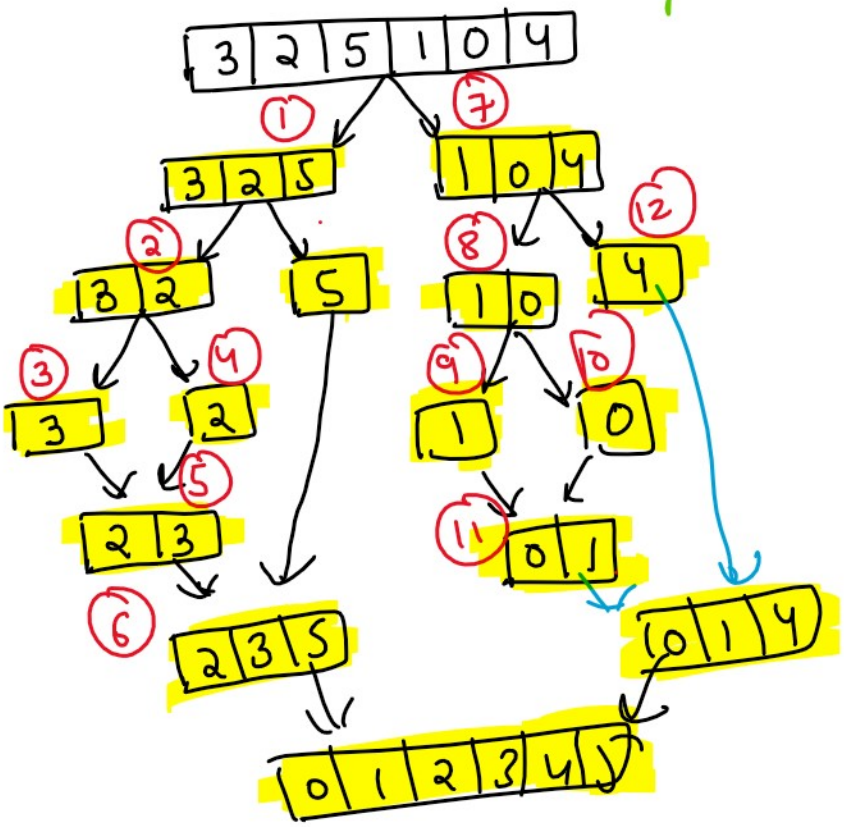
odd return a * power(a, b/2) * power(a, b/2);

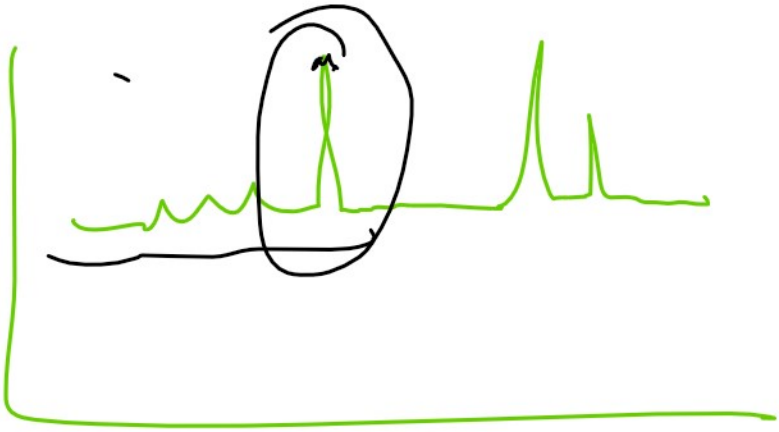
$5^{11} \rightarrow 5^{5} * 5^{5} * 5^{1}$

Merge Sort

20 August 2023 14:11

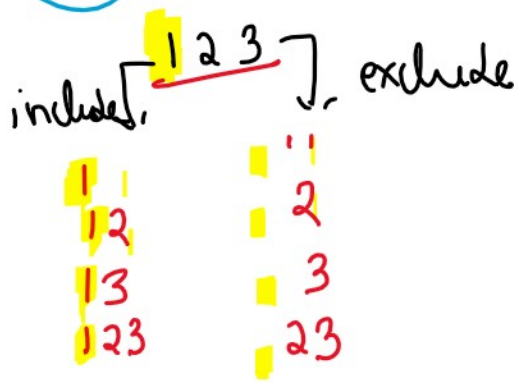



3 2 5 1 0 4
1
7
3 2 5
1 0 4
2
8
12
3 2
5
1 0
4
3
4
9
10
3
2
1
0
5
11
2 3
0 1
6
2 3 5
0 1 4
0 1 2 3 4 5



Subsets

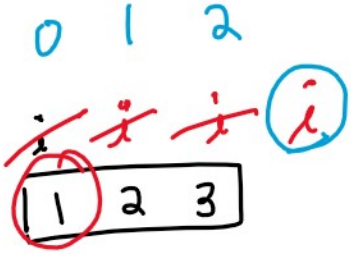
20 August 2023 15:03



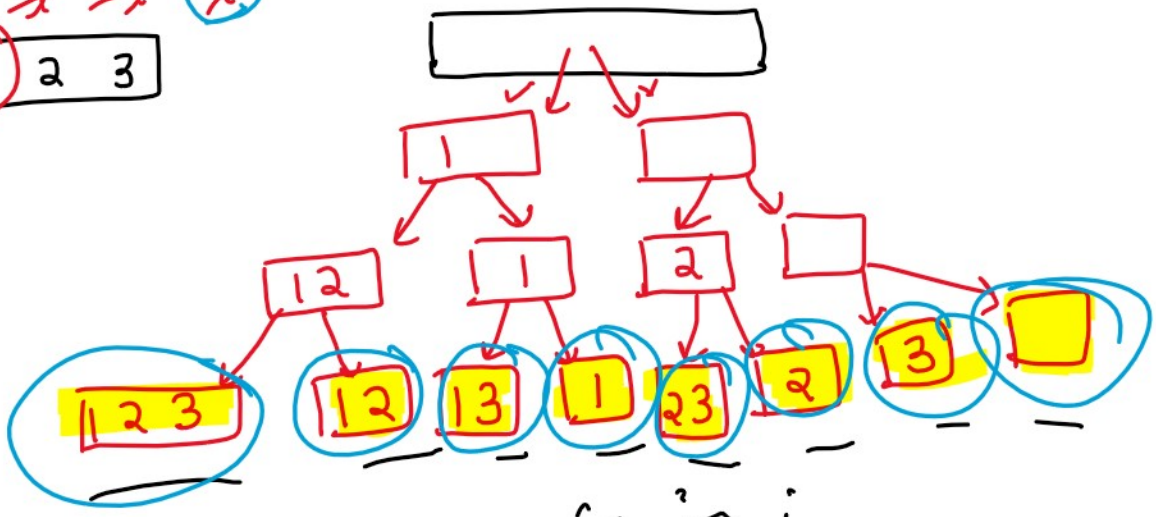
Base case ▸

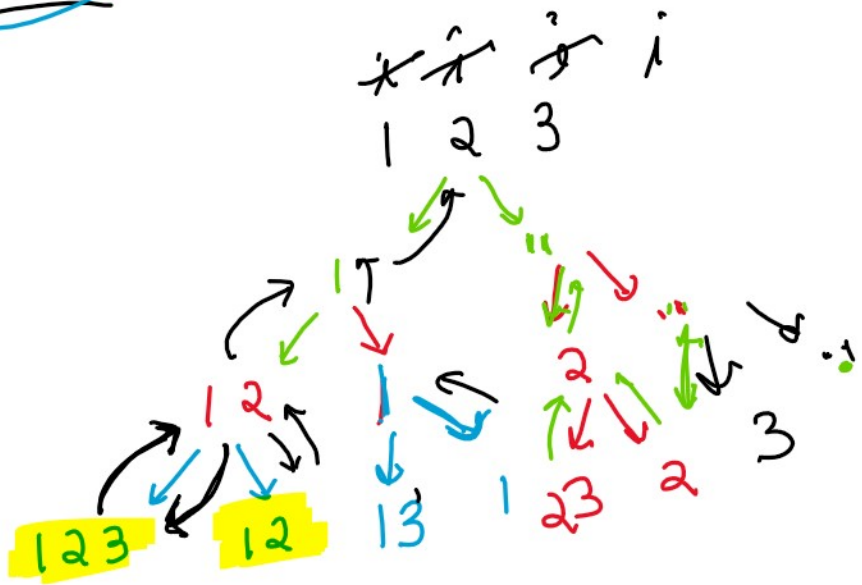
index == n

add in Ans.



j == 3





res

123

12

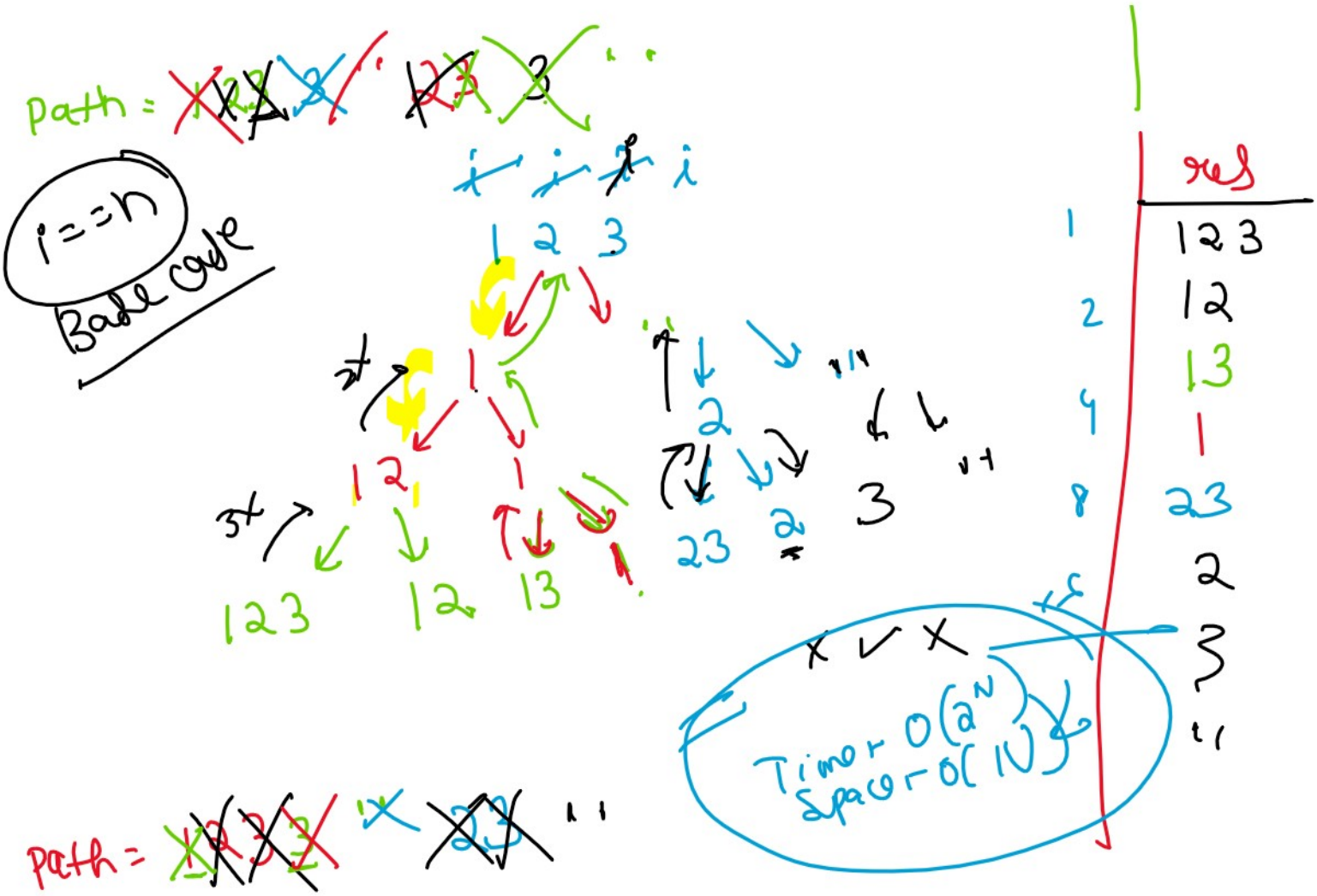
13

1

23

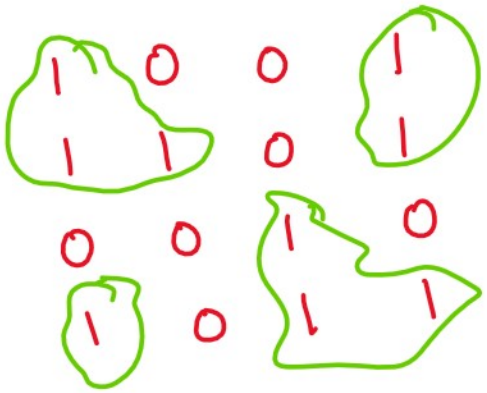
2

3


Path =
i == n
Base case
res
123
12
13
1
23
2
3
Time = O(2^N)
Space = O(N)
Path =

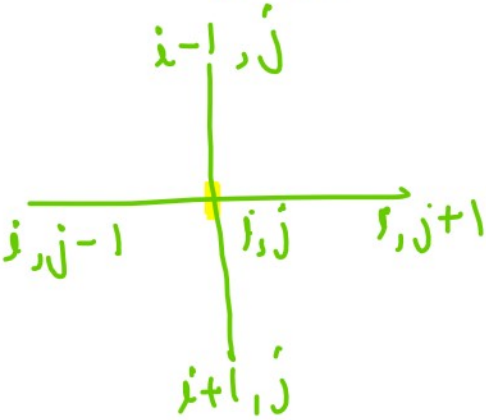
Number of islands

20 August 2023 15:48



1 → land

0 → water

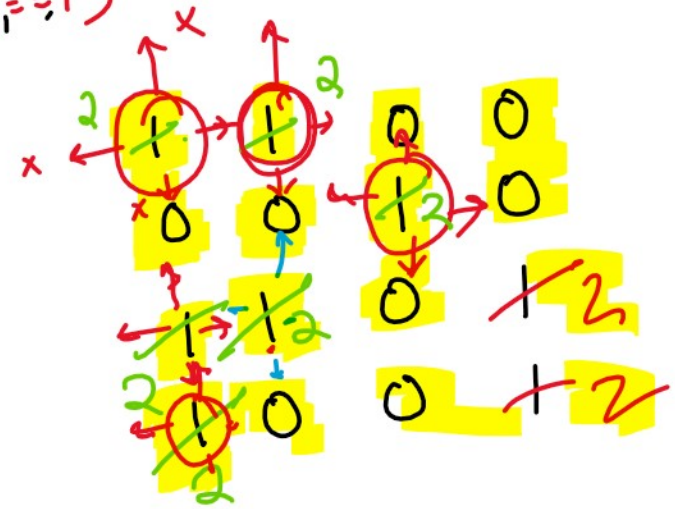


ans = 4

```
for(i=0; i<N) -> rows
    for(j=0; j<M) -> cols.
        if(grid[i][j]==1)
            //count++
```

NXM

dfs()



i=0, j=0 1

ans = 4

count = 0 1 2 3 4

1 → land

0 → water

2 → visited

Boundary check / base case

if i, j should be within grid

Time :- O(NXM)

Space :- O(NXM)

Time : -

Space - $O(N \times M)$

- ✓ → flood fill algo
- ✓ → max area island
- ✓ → surrounded region
- ✓ → word search (i, j)