Recursion & Backtracking

Date 17/06/
Page No.

fn calls itself directly or indirectly
↓
(recursive fn.)

```
static void recurse() {
        recurse()
    }
```

Recurse call

[Stackoverflow]
↓
error
recursive causes
filling of stack
memory

3 steps of recursion [to not cause & infinite running program]

→ find ~~best~~ base case
→ find relation between problem & subproblem
→ generalised this relation

1. Base case: when to leave the loop of recursion.
2. Subproblem: thing of subproblem.
   if P_1 & P_2 given is it possible to get P

$$\begin{tikzcd} & P \arrow[dl] \arrow[dr] & \\ P_1 & & P_2 \end{tikzcd}$$

Print N natural numbers

1. base case => n=0

```
if n==0{
    return n;
```

$$\begin{tikzcd} f^n(n) \arrow[d] \\ f^n(n-1) \end{tikzcd}$$

Step 2:

1. If there is a sub problem
2. do this subproblem have a sub-structure [similar structure?]

if this 2 exist then can be solved recursively.

Anuj's method: take some example

$f^n(5) = "5" + f^n(4)$ ["Recursive leap of faith"]

↓

if i get 4 i can solve 5 this should be faith

don't go deep in recursion just go with flow
↓
if P can be solved using P_1 and P_2 don't go under P_{11}, P_{12} P_{21}, P_{22}

generalising

$f^n(n) = "n" + f^n(n-1)$

```
fn(n){
    print(n)
    fn(n-1)
}
```

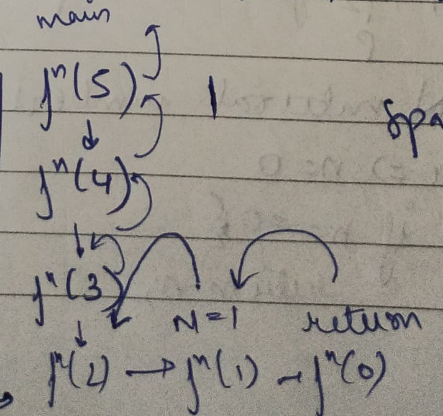
5, 4, 3, 2, 1, ~~0~~
↓
return

↑

Tail recursion: after recursive call fn there is no statement that is Tail recursion

```
fn(n)
    if(n==0) return
        fn(n-1)
print(n);
```

O(n)



Space O(N)

Sum of n Natural Numbers

S1: Base Case: $f^n(0)$

multiple base case possible

$n==0$ return 0;

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

S2: relation

5+4+3+2+1+0

$\underbrace{\overset{n}{f}(4)}_{1+2+3+4} = \underbrace{\overset{n}{f}(3)}_{1+2+3} + \cancel{x}4$

S3: generalized

$f^n(5) \rightarrow f^n(4) + 5$
↓
don't know
don't think how

$f^n(n) = f^n(n-1) + n$

→

call 5 → call 4
1 - - - 4
{don't think how}

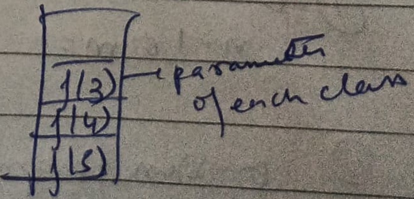
→ simpler & less time to write & debug

Advantages & Disadvantages

- new storage location for variables

→ slow

CP → no recursion



find power of a number

pow (a,b)

(3,4) → 3^4 }

S1: base case

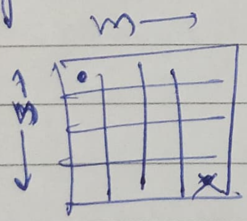
~~b~~=0 return 1;
a=0 return 0;
a=1 return 1;
b=1 return a;

S2: $3^4 = 3^3 * 3$

~~B~~ $f^n(n) = f^n(n-1) * n$;

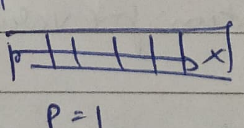
→ find number of paths in nxm Matrix:

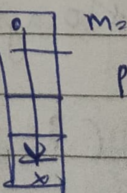
$O(2^n)$



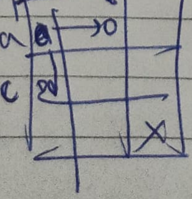
Rule: Right or down.
Start(0,0) end(n-1,m-1)

① base case
n=1 or m=1 return1;





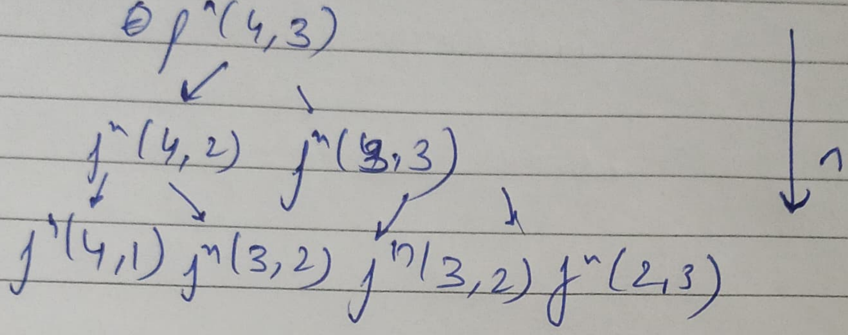
② problem / subproblem if i wanna



if i wanna go from a if sold from b that can help. & c

③ ~~$f^n(0,n$~~

$f^n(n,m) = f(n-1,m) + f(n,m-1)$ ~~$+2$~~



DP give $O(n+m)$