Recursion 1

Happy Makar Sakranti

AGENDA:

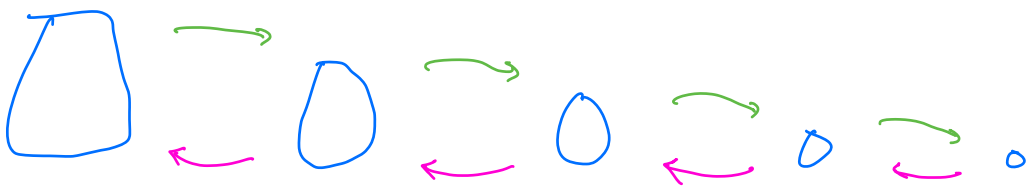
- What is recursion ?
- How to write recursion code ?
- How it works ?

What is Recursion ?

Function calling itself

Observations

1) Size keeps decreasing
2) Similar dolls
3) End doll



Solving a problem using a smaller instance of the same problem

subproblem

Example: Sum of first N natural numbers

N → Natural number
N > 0

Sum(N) → 1 + 2 + 3 + 4 ..... + N-1 + N

Sum of all no.s till N-1

Sum (N-1)

⇒ Sum (N) = Sum (N-1) + N ← Main Logic

Subproblem

Steps

1) Make an assumption
   ↳ Decide what your function does & trust that it will do it.

2) Main Logic
   ↳ Solve your bigger problem using a subproblem

3) Base case
   ↳ When your recursion stops

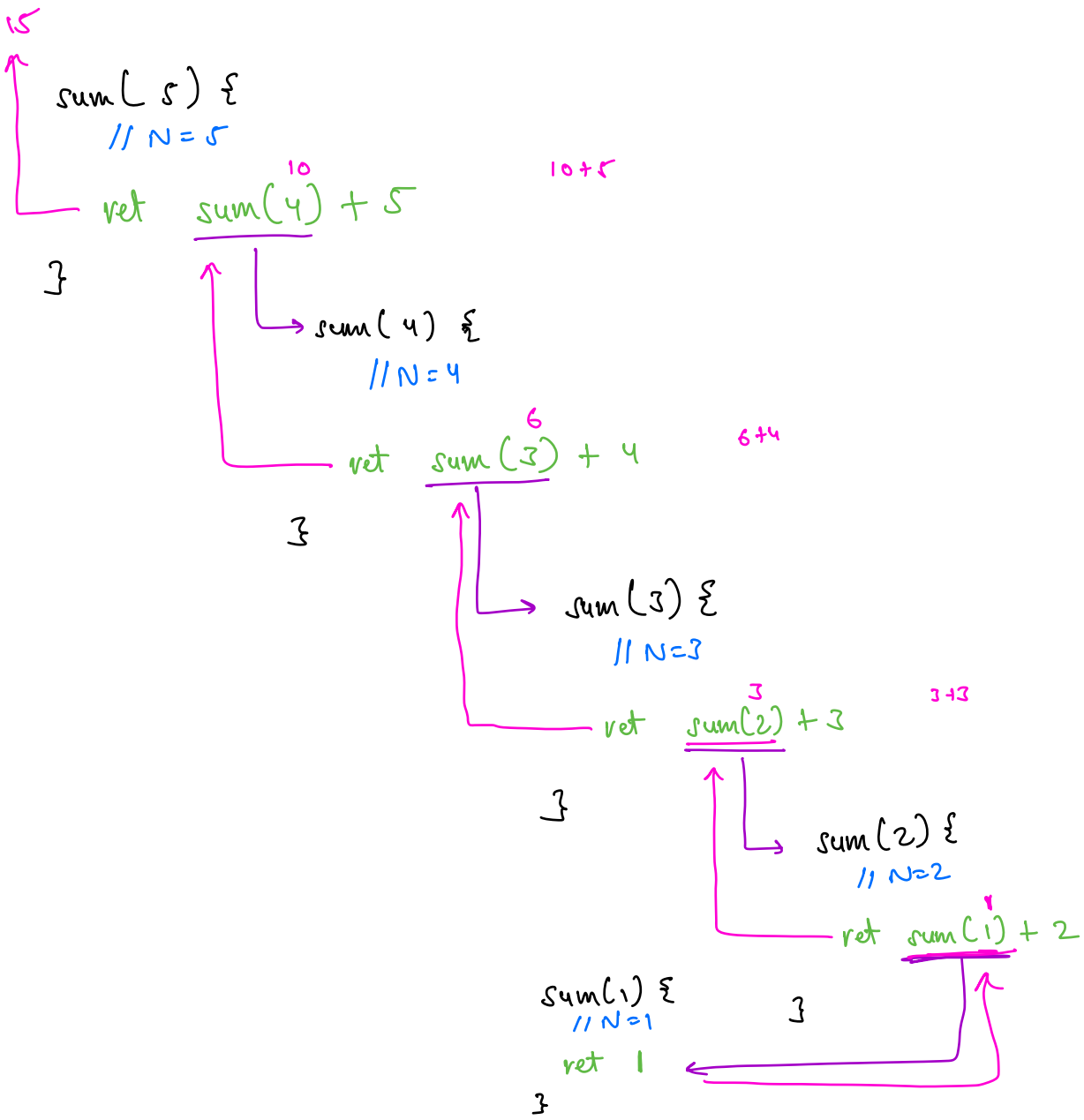
Assumption – Sum(N) gives us sum of all numbers from 1 to N.

```
sum ( int  N) {

    if (N==1)
        return 1          <- Base Case

    return  sum(N-1)  + N     <- Main Logic

}
```

Dry Run - sum(N)

```
sum ( int N) {
    if (N==1)
        return 1

    return sum(N-1) + N
}
```

N=5



Example: Factorial of N

N!

fact(N) = 1 x 2 x 3 x 4 ..... x(N-1) x N

(N-1)!

fact(N) = fact(N-1) x N

Subproblem

Quiz 1

0! = 1

1! = 1

Assumption - fact(N) gives us N!

```
fact ( int N) {
    if (N==0)          <- Base Case
        return 1

    return fact(N-1) x N   <- Main Logic
}
```

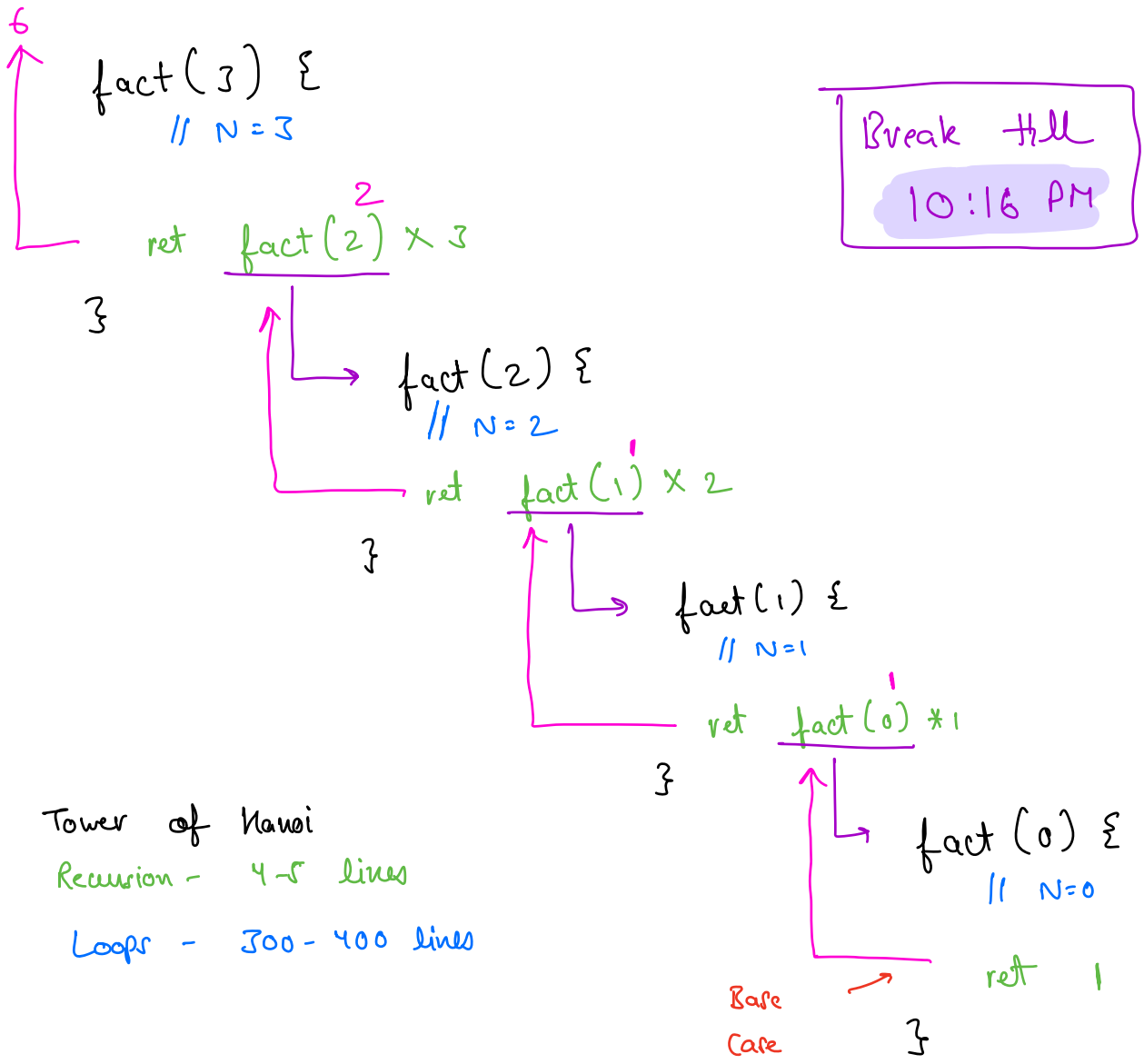
Dry Run - Factorial

```
fact ( int N) {
    if (N==0)
        return 1
    return fact(N-1) x N
}
```

N = 3

↓

Ans = 6



Break till 10:16 PM

Tower of Hanoi

Recursion - 4-5 lines

Loops - 300 - 400 lines

Example: Fibonacci Series

Golden Ratio

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

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 ------

Given N, compute N^{th} fibonacci number

$$fib(N) = fib(N-1) + fib(N-2)$$

if N == 0
fib(0) = ~~fib(-1)~~ + ~~fib(-2)~~
return 1

if N == 1
fib(1) = fib(0) + ~~fib(-1)~~
return 1

Assumption – fib(N) gives us N^{th} fibonacci no.

```
fib ( int N) {
    if (N == 0 or N == 1)        ← Base Case
        return 1

    return fib(N-1) + fib(N-2)   ← Main Logic
}
```

Dry Run – Try on your own

Q1 Given a number N, print all numbers from 1 to N in increasing order using recursion.

incPrint(5) → 1, 2, 3, 4, 5
incPrint(4)
print(5)

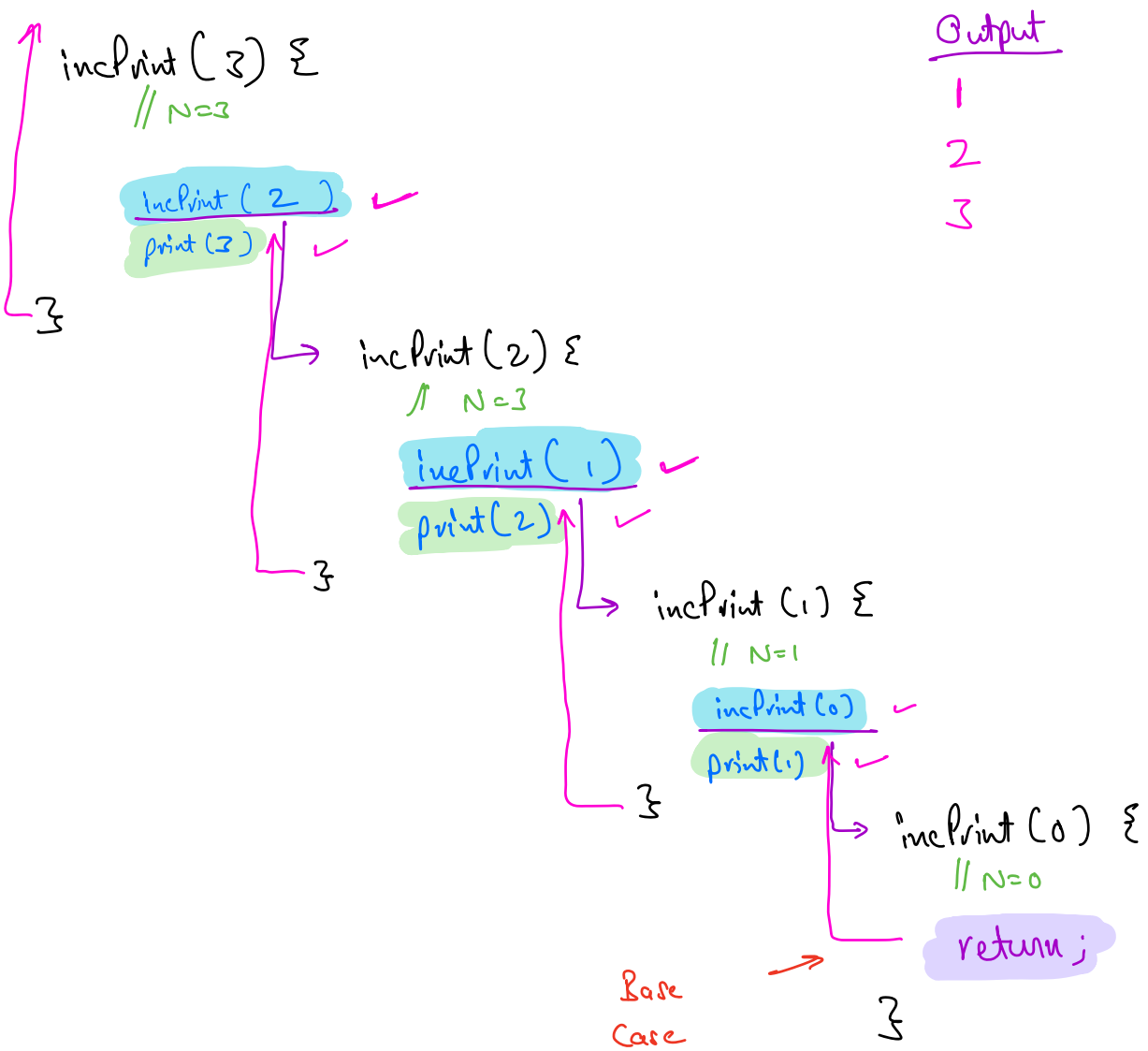
incPrint(N)
↳ incPrint(N-1)
print(N)

Assumption — incPrint(N) will print all nos from 1 to N

```
incPrint(int N) {
    if (N==0) {
        return
    }
    incPrint(N-1)
    print(N)
}
```

Base Case

Main Logic


incPrint(3) {
// N=3
incPrint(2)
print(3)
}
incPrint(2) {
// N=3
incPrint(1)
print(2)
}
incPrint(1) {
// N=1
incPrint(0)
print(1)
}
incPrint(0) {
// N=0
return;
}
Base Case
Output
1
2
3

Q2 Given a number N, print all numbers from N to 1 in decreasing order using recursion.

decPrint(5) → 5, 4, 3, 2, 1

print(5)

decPrint(4)

TODO

One or two lines change in the previous code

Doubts

Thank You

- Mergesort
- Time Complexity

} Next Class

Problem → Broken down into a subproblem

Factors

1) Time
2) Space

} Loops win

In most cases

Not a rule

3) Simplicity — Recursion

Good Night

Thank You

Monday