Recursion Basics

Friday, 25 June 2021 5:08 PM

Recursion: (Function calling itself)

$sum(N) = 1+2+3+4+\cdots+N-1+N$

$= sum(N-1) + N$

$sum(N) = sum(N-1) + N$

[Solving a Problem, using it's subproblem
It is called as recursion]

1) **Assumption:** { Your function does what you want to do ✓ }

2) **Main logic:** Solving function using subproblem ✓

3) **Base Condition:** Input value for which we should stop recursion.

```
int sum(int N){
① if(N==0) return 0
② return sum(N-1) + N
}
```

1) Assumption: { Given N, calculate sum of 1 to N natural numbers & return it. }

→ Main logic

⇒ sum(5) : 15

15 ↳ return [sum(4)] + 5

10 ↳ return [sum(3)]6 + 4

6 ↳ return [sum(2)] + 3

3 ↳ return [sum(1)] + 2

1 ↳ return [sum(0)]0+1 + 1

return 0

Fibonacci series

0 1 2 3 4 5 6 7 (8) 9 ..

0 1 1 2 3 5 8 13 21 34 ..

n-2 n-1 n

```
int fib(int N){
if(N==0) return 0
if(N==1) return 1
return fib(N-1) + fib(N-2)
}
```

Assumption: Return N^{th} fib number

Main logic:

N=2: → (N-1) (N-2)

1 0

Base Condition:

TC: $[2^N]$ ✓

N: 1, 2, 3, .. N

```
void pri(int N){
① if(N==0) return;
② pri(N-1) // → [1, → N-1]
③ print(N);
}
```

Recursion:

Ass: { Given (N) print all number from 1 → N }

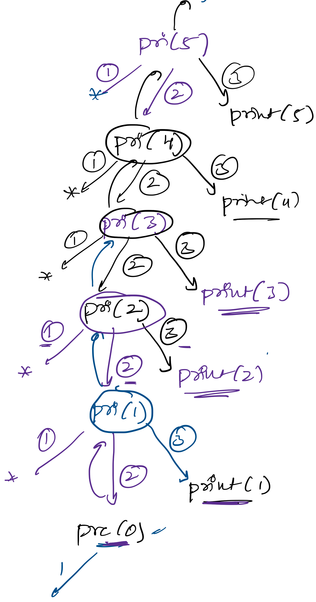
Main logic

Base Condition:

pri(5): 1 2 3 4 5

pri(7): 1 2 3 4 5 6 7

pri(N-1): 1 2 3 4 5 6 7 .. N-1



output: { 1, 2, 3, 4, 5 }

③ Given N, print numbers in desc N, N-1, .. 1

```
void pri(int N){
if(N==0) return;
print(N);
pri(N-1);
}
```

(N-1, N-2, .. 1)

Assumption: Given N, it will print N, N-1, N-2, .. 1

Main logic:

Base Condition:

pri(5): 5 4 3 2 1

pri(7): 7 6 5 4 3 2 1

pri(N-1): N-1, N-2, N-3 .. 1