Class 2.1 - Recursion

Saturday, May 17, 2025 10:51 AM

PMI

Principle of Mathematical Induction

231 → 23 + 221 → x => x

sub problems

smaller sub part / sub version

$f(\frac{n}{2})$ — f(0) true

— f(k)

— f(k+1)

Recursion

↳ function calling itself.

Parts of a Recursive solution

→ Base case

→ Recursive call (man lo shi answer aayega)

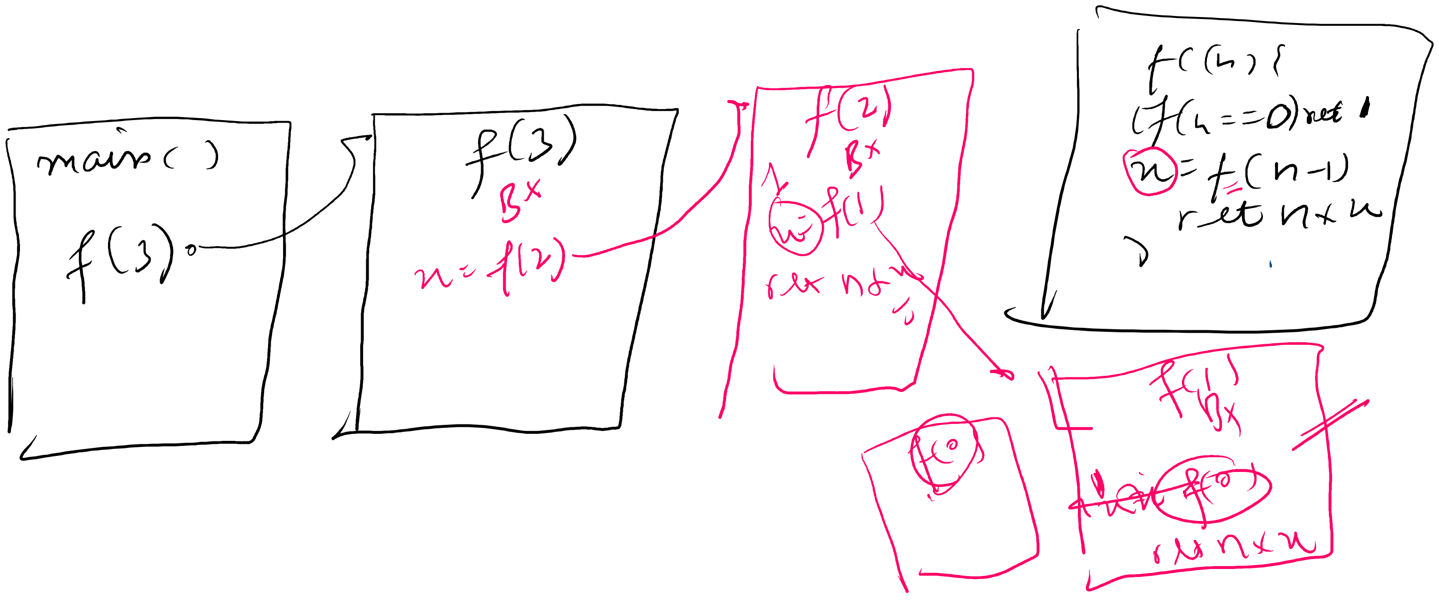
→ self work

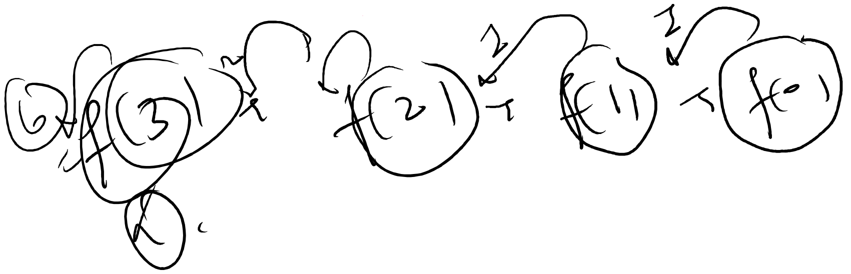
```
fact (n) {
  if(n==1) ret 1;

x = fact (n-1);
  ret n × x
}
```

→ Base case → The smallest case for which the answer is obvious)

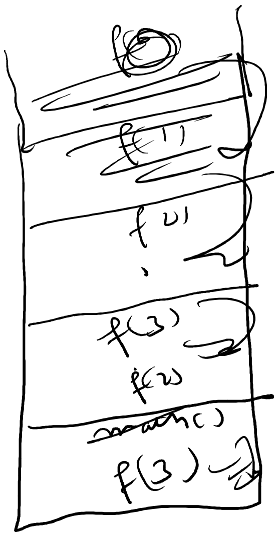
→ Base case → (The sm
which the answer is obvious /
already known.





⊛ Base case is used to terminate the recursion.

If not written / written incorrectly, you will end up in an infinite recursion loop & stack overflow error



Time - O(n)

S - O(n)

★ Base case is always written on the changing variable

fib

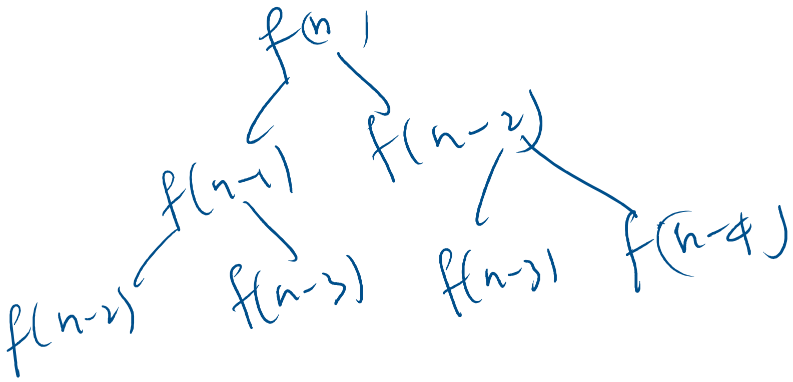
fib(n)
↓

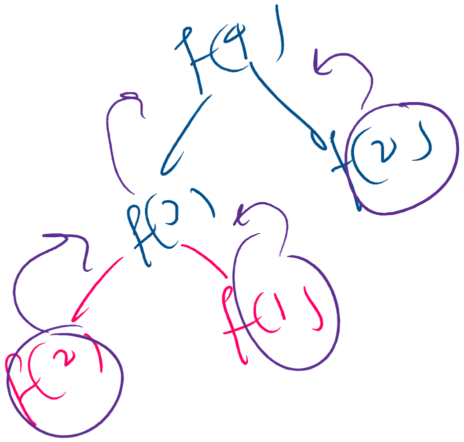
0 1 1 2 3 5 8

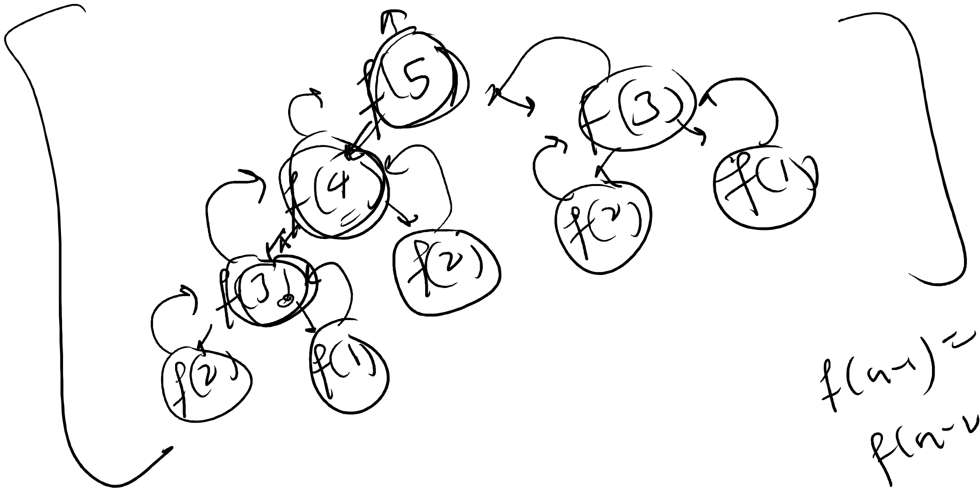
```
fib(n) {
    if(n==1) ret 0;
    if(n==2) ret 1;
    f1 = fib(n-1)
    f2 = fib(n-2)
    ret f1+f2
```

$f2 = 1$

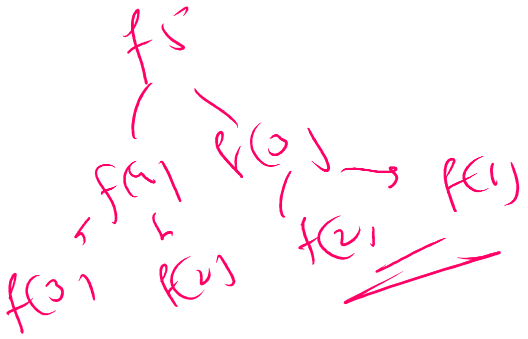
2 ret $= f1 + f2$

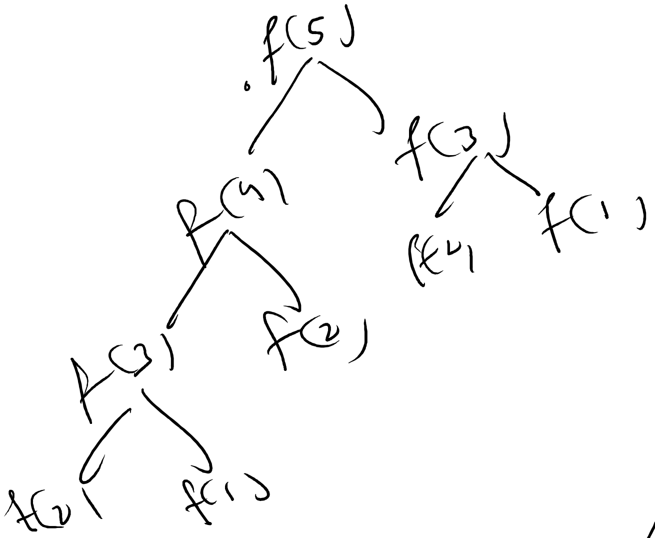


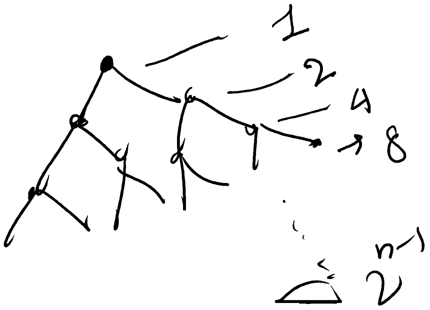




$f(n-1) =$
$f(n-2)$







⊛ Whenever there is branching in your recursion, use the formula to find T.C.

$$(\text{No. of Branches})^{\text{depth of recursion}}$$

(No. of Branches)