RECURSION QUESTIONS

FIBONACCI

0 1 1 2 3 5 8 . . .

$fib(0) = 0$; $fib(1) = 1$

$fib(2) = fib(1) + f(0)$

$= 1$ . . . .

HIGH-LEVEL THINKING

Expectation

n^{th} position tak ki fibonacci sequence lengy

fib(5) = 0 1 1 2 3 5

Faith

fib(n-1) = we can find

fib(4) = 0 1 1 2 3

fib(3) = 0 1 1 2

Expectation meets faith

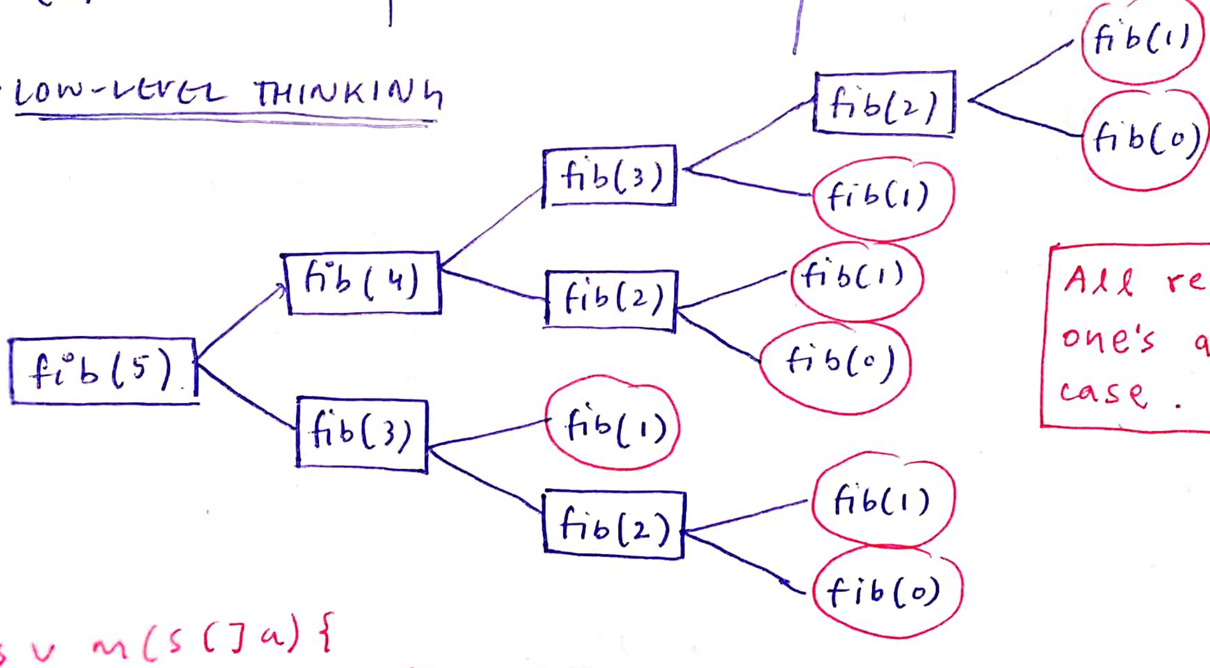
∴

$fib(5) = fib(4) + fib(3)$

$= 3 + 2$

$= 5$

LOW-LEVEL THINKING



All red circle one's are base case.

```
p s v m(S[] a){
Scanner s = new Scanner(System.in);
int n = s.nextInt();
int ans = fib(n);
syso(ans);
}
```

```
p s int fib(int n){
 if(n==0||n==1){ //Base Case
    return n;
 }
int fibnm1 = fib(n-1); ]Recursive
int fibnm2 = fib(n-2); ] calls
int fibn = fibnm1 + fibnm2;
return fibn;
}
```

T(n-1)

T(n-2)

K

Time complexity

$f(2) = f(1) + f(0)$
$f(3) = f(2) + f(1)$
$f(4) = f(3) + f(2)$
$\vdots$

$$f(n) = \underbrace{\underbrace{f(n-1)}_{T(n-1)} + \underbrace{f(n-2)}_{T(n-2)}}_{K}$$

$T(n) = T(n-1) + T(n-2) + k$

$\therefore \{T(n-2) = T(n-1) - T(1)\}$

$T(n) = T(n-1) + T(n-1) - T(1) + K$

$T(n) + T(1) = T(n-1) + T(n-1) + K$

Agar hume $T(1)$ ko ignore karna hai!

$\boxed{T(n) + T(1)}$ equal hai $\boxed{T(n-1) + T(n-1) + K}$ ke!

$\therefore$ Toh sirf $T(n)$ kam hoga $\boxed{T(n-1) + T(n-1) + K}$ se!

$\therefore T(n) \leq T(n-1) + T(n-1) + K$

$\boxed{T(n) \leq 2T(n-1) + K}$ – Initial equation

$\therefore$

$(T(n) \leq 2T(n-1) + K)\, 2^0$
$(T(n-1) \leq 2T(n-2) + K)\, 2^1$
$(T(n-2) \leq 2T(n-3) + K)\, 2^2$
$\vdots$
$(T(1) \leq 2T(0) + K)\, 2^{n-1}$

$\Downarrow$

$T(n) \leq K + 2\cancel{T(n-1)}$
$2\cancel{T(n-1)} \leq 2K + 4\cancel{T(n-2)}$
$4\cancel{T(n-2)} \leq 4K + 8\cancel{T(n-3)}$
$8\cancel{T(n-3)} \leq 8K + 16\cancel{T(n-4)}$
$\vdots$
$2^{n-1}\cancel{T(1)} \leq 2^{n-1}(K)$

$T(n) \leq K + 2K + 4K + \dots 2^{n-1}K$

$T(n) \leq K(\underbrace{2^0 + 2^1 + 2^2 + \dots + 2^{n-1}}_{GP})$

$T(n) \leq K\left[\text{Sum of GP} = \frac{a(r^n - 1)}{r - 1} \quad a = 2 \quad r = 1\right]$

$$T(n) \leq K\left(1 \frac{(2^n - 1)}{2 - 1}\right)$$

$$T(n) \leq \frac{K(2^n - 1)}{1}$$

$$T(n) \leq K(2^n) + K$$

$$T(n) \leq 2^n K - K$$

$$\boxed{T(n) \propto 2^n} \Rightarrow \boxed{T(n) = O(2^n)}$$

☆ Hume ek FOR LOOP given hai (it is not a nested loop)
∴ It's Time Complexity is $O(n^2)$. WHY?

```
For(int i=1, j=1; i<=n; j++)
{
   //do some work (k)
   if(j>i){
      j=1;
      i++;
   }
}
```

Let, $\boxed{n=4}$

☆ for $i=1$, loop (1) baar chalega!

∴ $\boxed{i=1,\ j=1,\ work=k}$

☆ for $i=2$, loop (2) baar chalega!

∴ i=2, j=1, work=K
i=2, j=2, work=K

☆ For i=3, loop (3) baar chalega!

i=3, j=1, work=k
i=3, j=2, work=k
i=3, j=3, work=k

☆ for i=4, loop (4) baar chalega!

i=4, j=1, work=k
i=4, j=2, work=k
i=4, j=3, work=k
i=4, j=4, work=k

∴ Time Complexity = $O(n^2)$

∴ Time Complexity

$n=4$

∴ $T(n) = K + 2K + 3K + 4K$

For (n) ∴ $T(n) = (K + 2K + \dots + nK)$

$T(n) = K(1+2+\dots n)$

$T(n) = \frac{K(n)(n+1)}{2}$

∴ $\boxed{T(n) \propto n^2}$

☆ Calculate Polynomial

Hume (x) aur (n) ki value given hai aur polynomial ki value nikalani hai using Formula!

Polynomial $= 1\cdot x^n + 2\cdot x^{n-1} + 3\cdot x^{n-2} + \dots \dots n\cdot x^1$

Hume vo function likhna hai jo polynomial ki value nikalde!

Example : $x=10$, $n=3$

Polynomial $(x,n) \longrightarrow$ Polynomial $(10,3)$

Polynomial $= 1\cdot x^3 + 2\cdot x^2 + 3\cdot x^1$

$= 1\cdot 10^3 + 2\cdot 10^2 + 3\cdot 10^1$

$= 1\cdot 1000 + 2\cdot 100 + 3\cdot 10$

$= 1000 + 200 + 30$

$= 1230$ Ans

Hume is function ko O(n) me hi solve krna hai!
HOW?
↳ function me LOOP ko REVERSE chalana padega!

$1 \cdot x^n + 2 \cdot x^{n-1} + \dots \dots + (n-2)x^3 + (n-1)x^2 + n \cdot x^1$

LOOP
REVERSED

$n x^{1} + (n-1)x^{2} + \dots \dots + (2)x^{n-1} + (1)x^n$

coefficent

Power of x

∴

```
c = coefficent  //coefficent
pox = x         //power of x
a = 0           //answer

while (c>=1) {
  term = c*pox;
  ans = ans + term;
  c--;
  pox = pox * x;
} //CODE SNIPPET
```

```
p s v m(s[]a){
scanner s = new scanner(system.in);
int x = s.nextInt();
int n = s.nextInt();
int c = n;
int pox = x;
int ans = 0;
while (c>=1){
 int term = c*pox;
 ans = ans + term;
 c--;
 pox = pox * x;
}
syso(ans);
}
```

→ Code tab tak chalega c→1 ho ya use bada ho!
means $1 \cdot x^n$ tak chalega

→ term = $n \cdot x^i$ → power of x
n ↓ coefficent

↳ sum total terms

↳ Decreased kuki coeffient will go from (n, n-1, ... 1.)