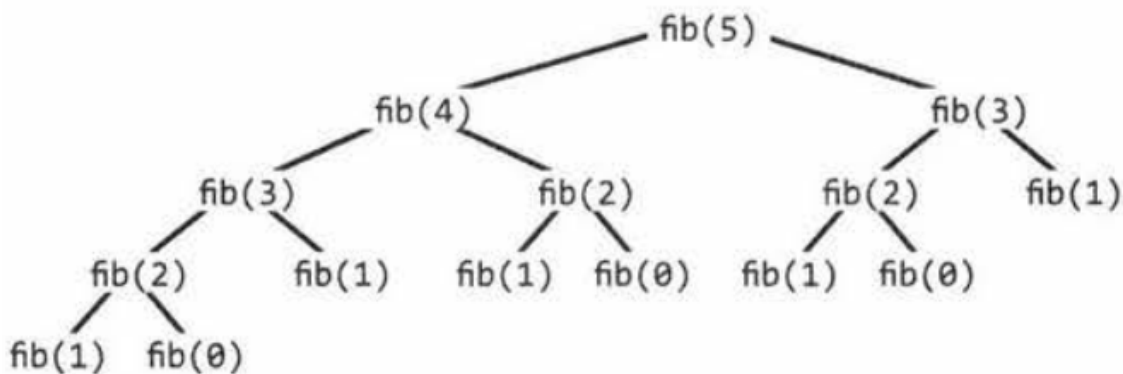


DP es un estilo de programación donde se usa recursión y memorización como recursos principales por ejemplo :

-Codigo recursivo para calcular un numero fibonacci, este algoritmo se aproxima a una complejidad $O(2^n)$:

```
int fibonacci(int i) {  
    if (i == 0) return 0;  
    if (i == 1) return 1;  
    return fibonacci(i - 1) + fibonacci(i - 2);  
}
```

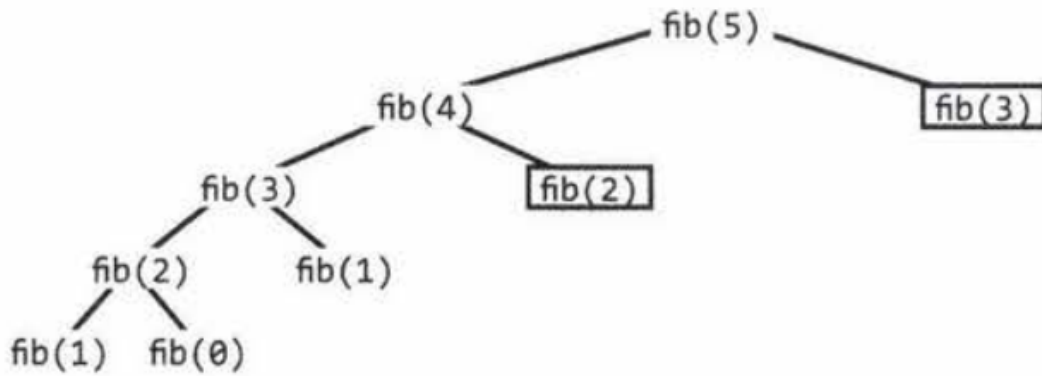
su arbol de llamadas es:



Como se puede observar muchas llamadas se repiten y se vuelven a recalcular, tecnicamente solo se deberían de computar n valores (5,4,3,2,1,0) una vez , memorizarlos y mandarlos a traer cuando sea necesario de tal manera que la complejidad del algoritmo se reduce a $O(n)$.

De la forma TOP-DOWN se van calculando los primeros valores y al final se sale de la recursion ya en los valores finales.

```
int fibonacci(int n) {  
    return fibonacci(n, new int[n + 1]);  
}  
  
int fibonacci(int i, int[] memo) {  
    if (i == 0 || i == 1) return i;  
  
    if (memo[i] == 0) {  
        memo[i] = fibonacci(i - 1, memo) + fibonacci(i - 2, memo);  
    }  
    return memo[i];  
}
```



De la forma BOTTOM-UP se calculan primero los valores mas bajos que rompen la recursion y con ese valor se van calculando los consecuentes:

```
int fibonacci(int n) {  
    if (n == 0) return 0;  
    else if (n == 1) return 1;  
  
    int[] memo = new int[n];  
    memo[0] = 0;  
    memo[1] = 1;  
    for (int i = 2; i < n; i++) {  
        memo[i] = memo[i - 1] + memo[i - 2];  
    }  
    return memo[n - 1] + memo[n - 2];  
}
```