

- La principal diferencia entre memoization y tabulation es que dado un problema, en memoization se usa recursión hacia los subproblemas mientras estos se resuelven de forma recursiva, almacenando los valores obtenidos en una estructura. Este es un approach Top-Down ya que va desde el problema original hacia los subproblemas más pequeños que el original. Tabulation es un approach iterativo que busca justamente evitar la recursión invirtiendo la forma en cómo se resuelven los problemas, este usa un Bottom-up approach, es decir, parte de él/los casos base y llega a la resolución del problema original de forma iterativa. La recursión tiene un overhead que es la que busca evitar tabulation, ya que esto la hace más lenta, por ende, **tabulation es un método más rápido que memoization, a pesar de que ambos tengan un time complexity de $O(n)$** . Como ejemplo se muestra el pseudocódigo de fibonacci para apreciar la diferencia entre ambos métodos de resolución:

- Memoization

```
fib(n, dp):  
    if dp[n] != null:  
        return dp[n]  
    if n==0 or n==1:  
        dp[n]=n  
        return n  
    else:  
        result = fib(n-1,dp) + fib(n-2,dp)  
        dp[n]=result  
        return result
```

- Tabulation

```
fib_tab(n):  
    if n==0 or n==1:  
        return n  
    dp= array of size n +1  
    dp[0]=0  
    dp[1]=1  
    for i=2 to n:  
        dp[i]=dp[i-1] + dp[i-2]  
    return dp[n]
```

- Las dos condiciones para poder usar programación dinámica en la resolución de un problema es que el problema cumpla lo siguiente:
 - Subproblemas superpuestos (overlapping):
 - Subestructura óptima: Entiéndase como que el problema se resuelve de forma óptima si y sólo si, los subproblemas que contiene también son resueltos de forma óptima por el mismo algoritmo.