

Memorization Programming

ปกติโค้ดสำหรับแสดงลำดับ fibonacci แบบ recursive จะเป็น

```
function fibo(n)
    if(n==0 or n==1)
        return n
    return fibo(n - 1) + fibo(n - 2)
```

ถ้าเราจะหาลำดับของ Fibonacci ลำดับที่ 5 เราต้องหา

$$\text{fibo}(5) = \text{fibo}(4) + \text{fibo}(3)$$

$$\text{fibo}(4) = \text{fibo}(3) + \text{fibo}(2)$$

$$\text{fibo}(3) = \text{fibo}(2) + \text{fibo}(1)$$

$$\text{fibo}(2) = \text{fibo}(1) + \text{fibo}(0)$$

$$\begin{aligned}\text{ดังนั้น } \text{fibo}(5) &= \text{fibo}(4) + \text{fibo}(3) \\ &= \text{fibo}(3) + \text{fibo}(2) + \text{fibo}(2) + \text{fibo}(1) \\ &= \text{fibo}(2) + \text{fibo}(1) + \text{fibo}(1) + \text{fibo}(0) + \text{fibo}(1) + \text{fibo}(0) + \text{fibo}(1) \\ &= \text{fibo}(1) + \text{fibo}(0) + \text{fibo}(1) + \text{fibo}(1) + \text{fibo}(0) + \text{fibo}(1) + \text{fibo}(0) + \text{fibo}(1) \\ &= 5\end{aligned}$$

จะเห็นได้ว่าจะเกิดการ call function fibo ซ้ำด้วยค่าเดิมที่เคยคำนวณมาแล้วหลายรอบ จะดีกว่าไหมถ้าเราจำไว้ว่า fibo(n) ที่เคยคำนวณไปแล้วได้ค่าอะไรออกมา?

```
function fibo_optimized(n)
    if(mem[n] is exist) return mem[n]
    if(n==0 or n==1)
        mem[n] = n
    return mem[n]
    mem[n] = fibo(n - 1) + fibo(n - 2)
    return mem[n]
```

Ref: <https://medium.com/@gepphkat/dynamic-programming-v-s-memorization-e146c9cb8f27>