

例題：フィボナッチ数列

問題：フィボナッチ数列

- 数列の各要素が、前の要素とその前の要素の和となっているような数列を「フィボナッチ数列」と呼びます

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

- 数学的には、以下のように記述できます
 - $F_0 = 0$
 - $F_1 = 1$
 - $F_{n+2} = F_{n+1} + F_n \ (n \geq 0)$
- この数列の n 番目の要素を計算するプログラムを考えます

案1：再帰呼出しを用いた実装

- 前回学習した再帰呼出しを用いて、「定義通りに」実装すると、以下のようになりますね？
 - 1個前の要素と、2個前の要素を計算し、足し合わせます

```
def fib1(n):  
    if n <= 0:  
        return 0  
    if n == 1:  
        return 1  
    return fib1(n - 1) + fib1(n - 2)
```

案1の実行時間

- 入力を、5, 10, 15, 20, 25, 30, 35, 40 と変化させながら、実行時間を測ってみましょう
 - セルの先頭に %%time を表示すると、実行時間が表示されます
 - 課題に使用しますので、時間は記録しておいてください

```
%%time  
fib1(5)
```

最初は順調ですが、35あたりから急に時間がかかってきましたね？

案1はなぜ遅いのか？

- この方法は、なぜ時間がかかるのかを考えてみましょう
- 例えば、5番目の要素を計算するには…

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

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

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

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

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

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

$$\text{Fib}(1) = 1$$

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

$$\text{Fib}(1) = 1$$

$$\text{Fib}(1) = 1$$

$$\text{Fib}(0) = 0$$

$$\text{Fib}(1) = 1$$

$$\text{Fib}(0) = 0$$

$$\text{Fib}(1) = 1$$

$$\text{Fib}(0) = 0$$

同じ要素を何度も計算していますね。これが無駄の原因です。

案2：効率的な実装

- 前の計算結果を記録しておいて、その結果を再利用するようにしましょう
 - 一般的に「動的計画法」と呼ばれるアプローチです

```
def fib2(n):  
    lst = [0] * (n + 1)  
    lst[0] = 0  
    lst[1] = 1  
    for i in range(2, n + 1):  
        lst[i] = lst[i - 1] + lst[i - 2]  
    return lst[n]
```

案2の実行時間

- 入力を、5, 10, 15, 20, 25, 30, 35, 40 と変化させながら、実行時間を測ってみましょう
 - 一瞬で計算が完了してしまうので、ここでは1000回繰り返した平均時間を測ってみます
 - セルの先頭に %%time を表示すると、実行時間が表示されます
 - 課題に使いますので、時間は記録しておいてください

```
fib2(5)
```

```
%%time
```

```
for i in range(1000):  
    fib2(5)
```

案1と同じ答えが、すぐに計算できましたね？

案1 vs 案2

- それぞれプログラムを実行すると結果は同じですが、実行時間には大きな差が出ました

n	Fib(n)	Time of fib1 (s)	Time of fib2 (s)
5	5	7.15×10^{-6}	1.26×10^{-6}
10	55	3.48×10^{-5}	1.51×10^{-6}
15	610	3.01×10^{-4}	1.99×10^{-6}
20	6765	3.28×10^{-3}	2.68×10^{-6}
25	72025	3.05×10^{-2}	3.45×10^{-6}
30	832040	3.17×10^{-1}	3.82×10^{-6}
35	9227465	3.04×10^0	4.24×10^{-6}
40	102334155	3.37×10^1	5.33×10^{-6}

案1と案2の計算量は？

- 案1の計算量 $\rightarrow O\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$
 - n 番目の要素を計算するためにかかる時間は、 $\left(\frac{1+\sqrt{5}}{2}\right)^n$ に比例することが知られています
 - $\left(\frac{1+\sqrt{5}}{2}\right)^5$ を計算すると約 11.1 となりますが、実際に n が 5 増えると、実行時間が1桁増えています
- 案2の計算量 $\rightarrow O(n)$
 - n 番目の要素を計算するためにかかる時間は、 n に比例します
 - 実際にグラフを書いてみると、概ね比例しています