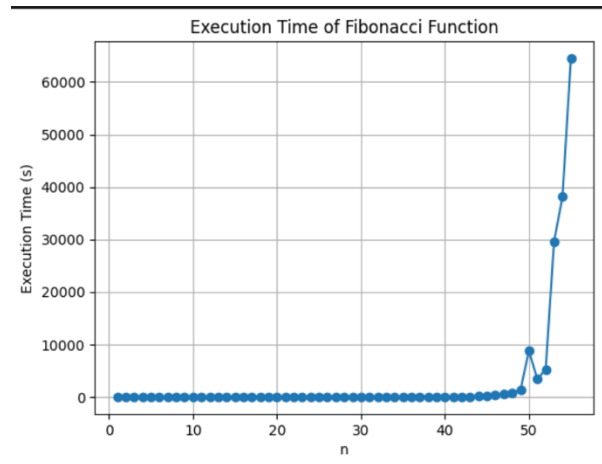


1. 編寫程式碼，使用兩種方法測量 $F(1)$ 、 $F(2)$ 、...、 $F(100)$ 的執行時間。將結果繪製為折線圖。

top-down (divide and conquer/ pure recursive) n 最大值為 55

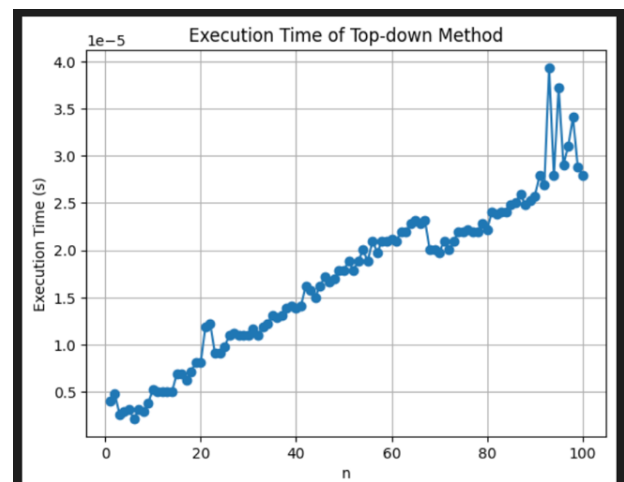
```
1 1.1920928955078125e-06
2 9.5367431640625e-07
3 9.5367431640625e-07
4 9.5367431640625e-07
5 9.5367431640625e-07
6 1.0073486328125e-06
7 3.0994415283203125e-06
8 3.814697265625e-06
9 6.9141387939453125e-06
10 1.0967254638671875e-05
11 1.5974044799804688e-05
12 2.7179718017578125e-05
13 4.315376281738281e-05
14 7.104873657226562e-05
15 0.00011420249938964844
16 0.00018310546875
17 0.0002970695495605469
18 0.0004951953887939453
19 0.0007879734039306641
20 0.0012540817260742188
21 0.0020258426666259766
22 0.003493785858154297
23 0.005292177200317383
24 0.008630097213745117
25 0.013881683349609375
...
52 5262.79190993309
53 29644.79679107666
54 38103.97762775421
55 64520.130469083786
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output
```



Maximum value of n reached: 55

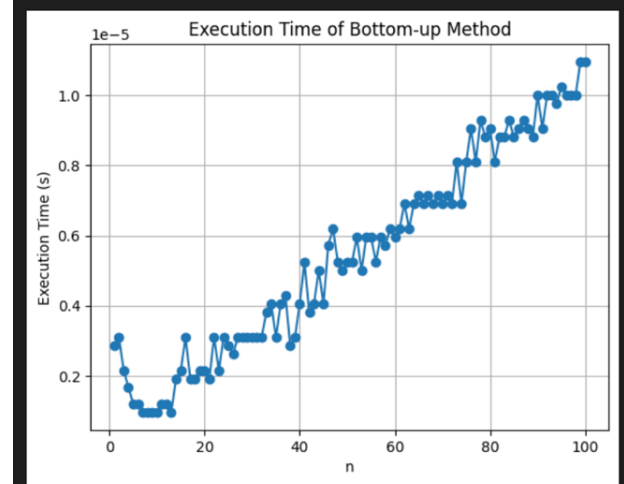
top-down (use cache)

多了一個 cache，然後從上而下遞迴從大問題開始解決，遇到重複子問題時，就從快取中取出答案，避免重複計算。



bottom-up (dynamic-programming-like)

用一個陣列來保存結果。然後在迴圈中將前面兩項的值相加就是目前項的解，然後又把這個解放入陣列中，又可以求出更後面的解。

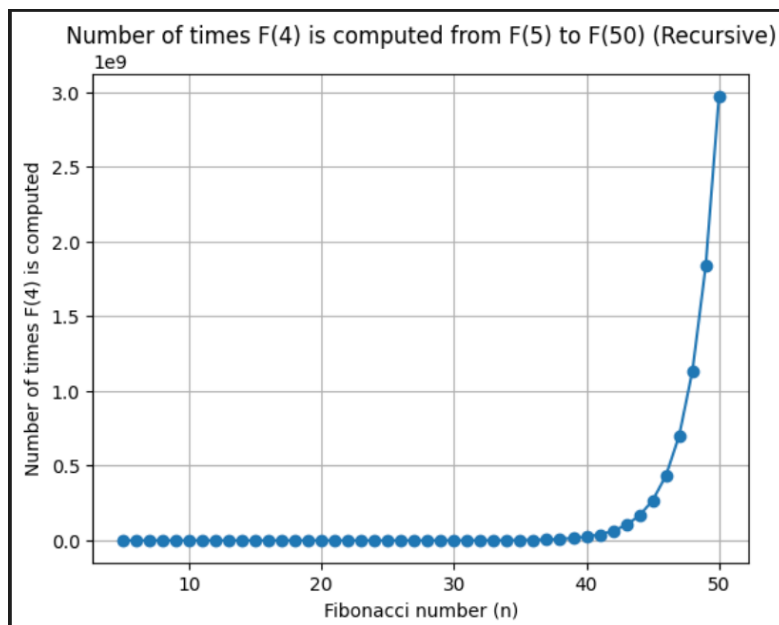


2.衡量子問題的重疊程度。執行 $F(5), F(6), \dots, F(50)$ 時計算 $F(4)$ 的次數。結果畫成折線圖。

top-down (divide and conquer/ pure recursive)

同一個子問題 $F(i)$ 會被計算很多次

```
計算 F(5) 的過程中計算 F(4) 的次數: 1
計算 F(6) 的過程中計算 F(4) 的次數: 2
計算 F(7) 的過程中計算 F(4) 的次數: 3
計算 F(8) 的過程中計算 F(4) 的次數: 5
計算 F(9) 的過程中計算 F(4) 的次數: 8
計算 F(10) 的過程中計算 F(4) 的次數: 13
計算 F(11) 的過程中計算 F(4) 的次數: 21
計算 F(12) 的過程中計算 F(4) 的次數: 34
計算 F(13) 的過程中計算 F(4) 的次數: 55
計算 F(14) 的過程中計算 F(4) 的次數: 89
計算 F(15) 的過程中計算 F(4) 的次數: 144
計算 F(16) 的過程中計算 F(4) 的次數: 233
計算 F(17) 的過程中計算 F(4) 的次數: 377
計算 F(18) 的過程中計算 F(4) 的次數: 610
計算 F(19) 的過程中計算 F(4) 的次數: 987
計算 F(20) 的過程中計算 F(4) 的次數: 1597
計算 F(21) 的過程中計算 F(4) 的次數: 2584
計算 F(22) 的過程中計算 F(4) 的次數: 4181
計算 F(23) 的過程中計算 F(4) 的次數: 6765
計算 F(24) 的過程中計算 F(4) 的次數: 10946
計算 F(25) 的過程中計算 F(4) 的次數: 17711
計算 F(26) 的過程中計算 F(4) 的次數: 28657
計算 F(27) 的過程中計算 F(4) 的次數: 46368
計算 F(28) 的過程中計算 F(4) 的次數: 75025
計算 F(29) 的過程中計算 F(4) 的次數: 121393
...
計算 F(47) 的過程中計算 F(4) 的次數: 701408733
計算 F(48) 的過程中計算 F(4) 的次數: 1134903170
計算 F(49) 的過程中計算 F(4) 的次數: 1836311903
計算 F(50) 的過程中計算 F(4) 的次數: 2971215073
```



bottom-up (dynamic-programming-like)

當我們計算費波那契數列時，每個數字都是由前兩個數字相加得到的（除了最初的幾個數字）。當我們計算 $F(5)$ 時，我們需要使用 $F(4)$ 和 $F(3)$ 。因此，計算 $F(5)$ 時， $F(4)$ 被計算了一次。

在之後的計算中，當我們計算 $F(6)$ 、 $F(7)$ 、...、 $F(50)$ 時，我們都不需要重新計算 $F(4)$ ，因為它已經在計算 $F(5)$ 時被計算過了。因此從 $F(5)$ 到 $F(50)$ 期間，每個數字中 $F(4)$ 只會被計算一次。

