遞迴

Recursion

2015 資訊之芽/語法班, Gin

介紹

- 定義：自己定義自己！
- 例子：有座廟，廟裡有個老和尚，正在給小和尚講故事呢！故事是什麼呢？「有座廟，廟裡有個老和尚，正在給小和尚講故事呢！故事是什麼呢？『有座廟，廟裡有個老和尚，正在給小和尚講故事呢！故事是什麼呢？……』
- 例子：兩面鏡子不停反射

數學上的定義

- 考慮等差數列 0, 1, 2, 3, 4, 5, 6, 7, 8
  其遞迴關係式為

  $b_{n+1} = b_n + 1, n \geq 0$

  $b_0 = 0$

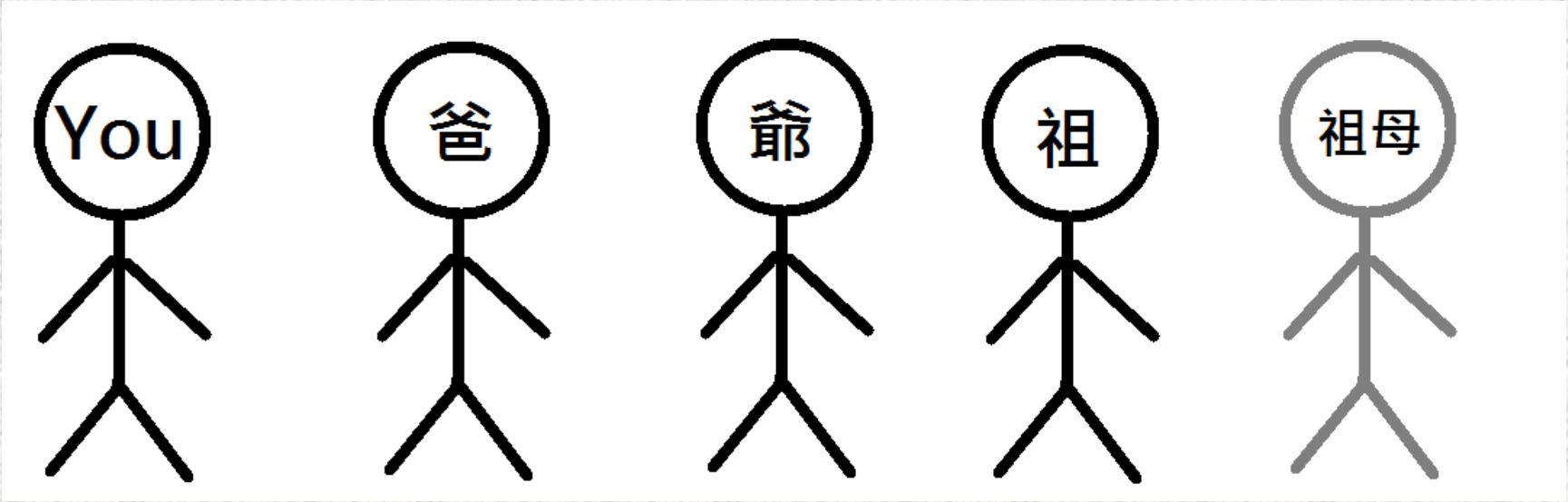
  每一項的數字等於前一項加上一

程式上的定義

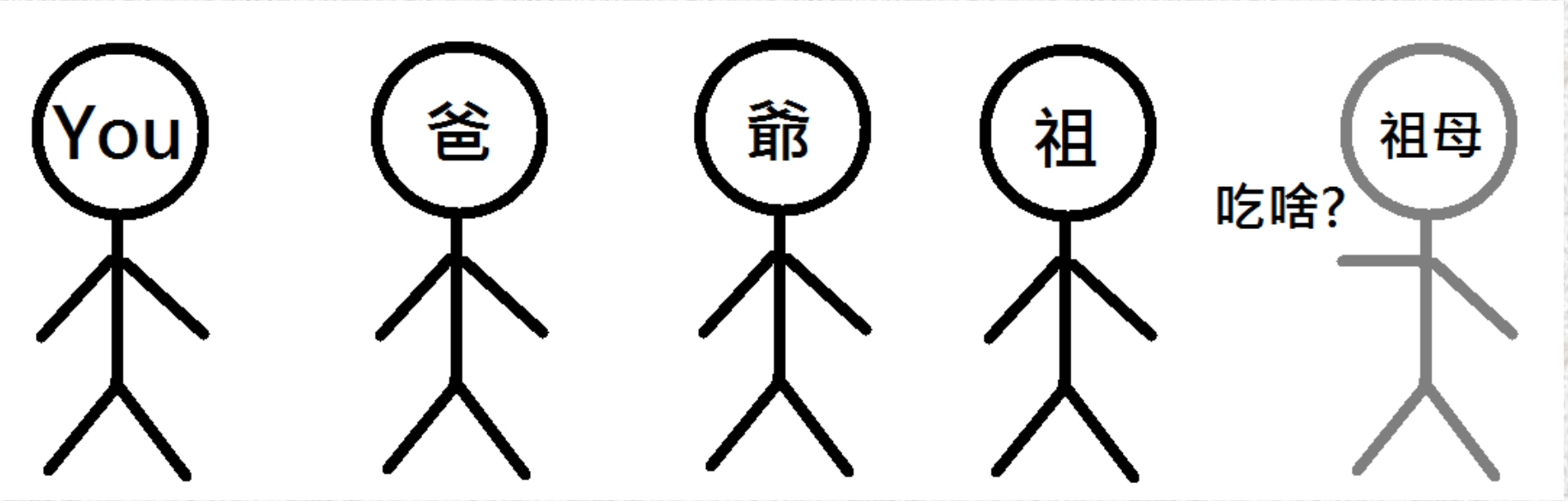
- 定義：回傳自己！
- ```
  int function( int n )
  {
      if( n == 1 )
              return 1;
      else
              return function( n - 1 );
  }
  ```
- 這裡看不懂沒關係，我們會一步一步說明
  ```

觀念

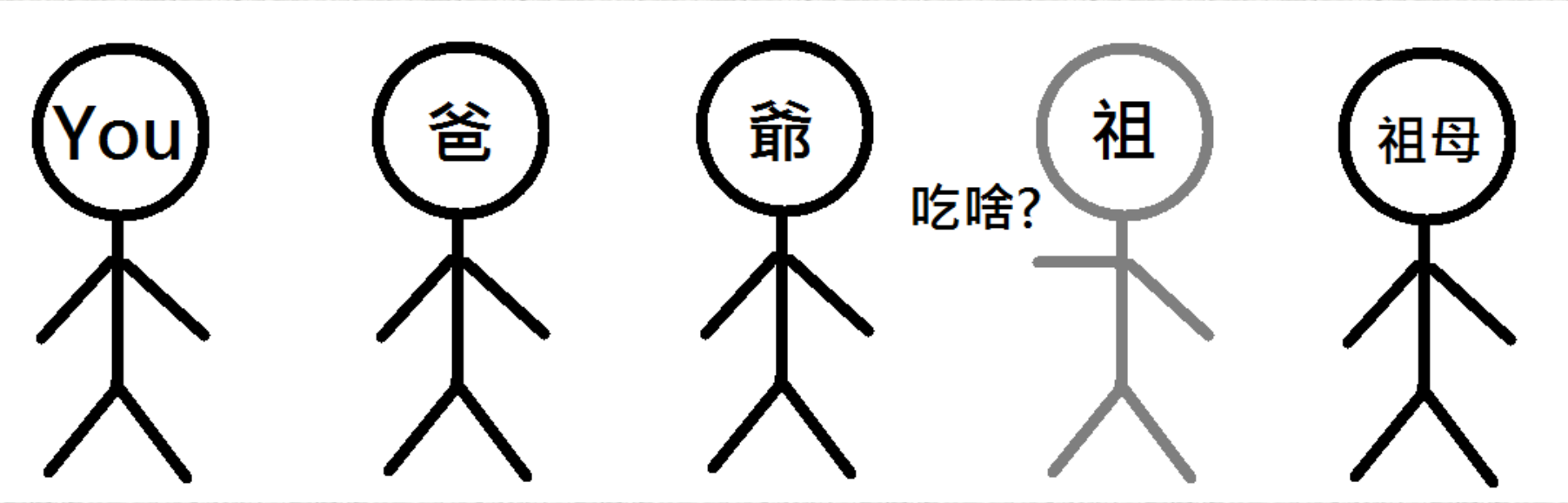
- 設想一個情景：
  你家有
  　　你/妳 & 爸爸 & 爺爺 & 祖父 & 祖母
  祖母問祖父晚上要吃什麼？



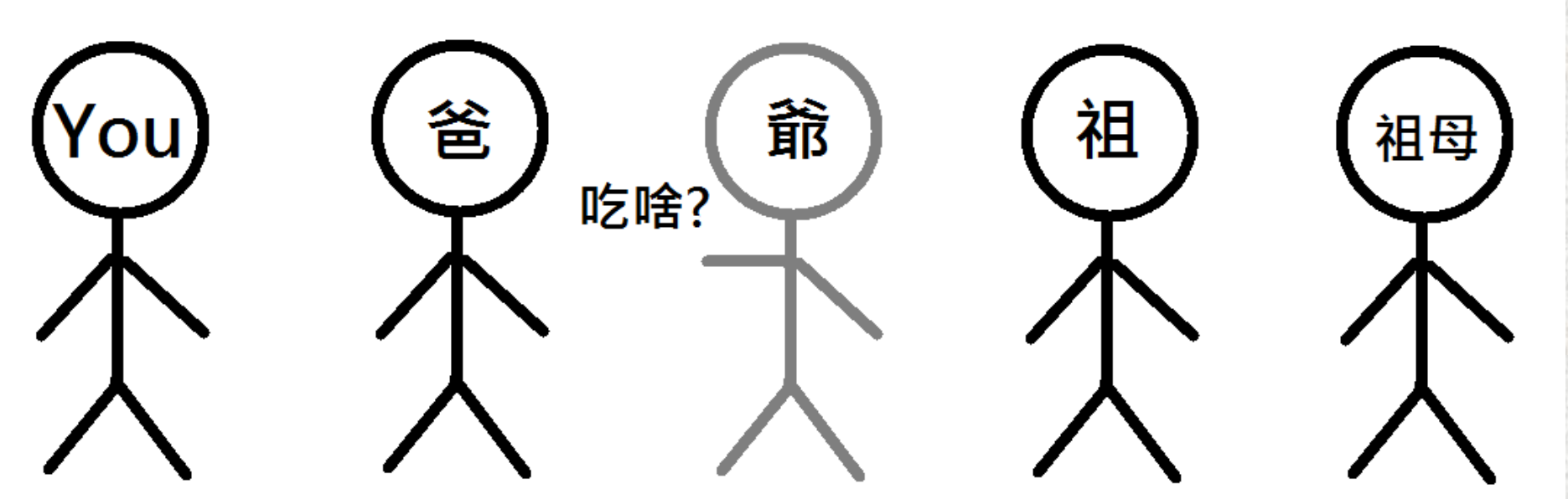
- 祖母問說晚上要吃什麼.....?



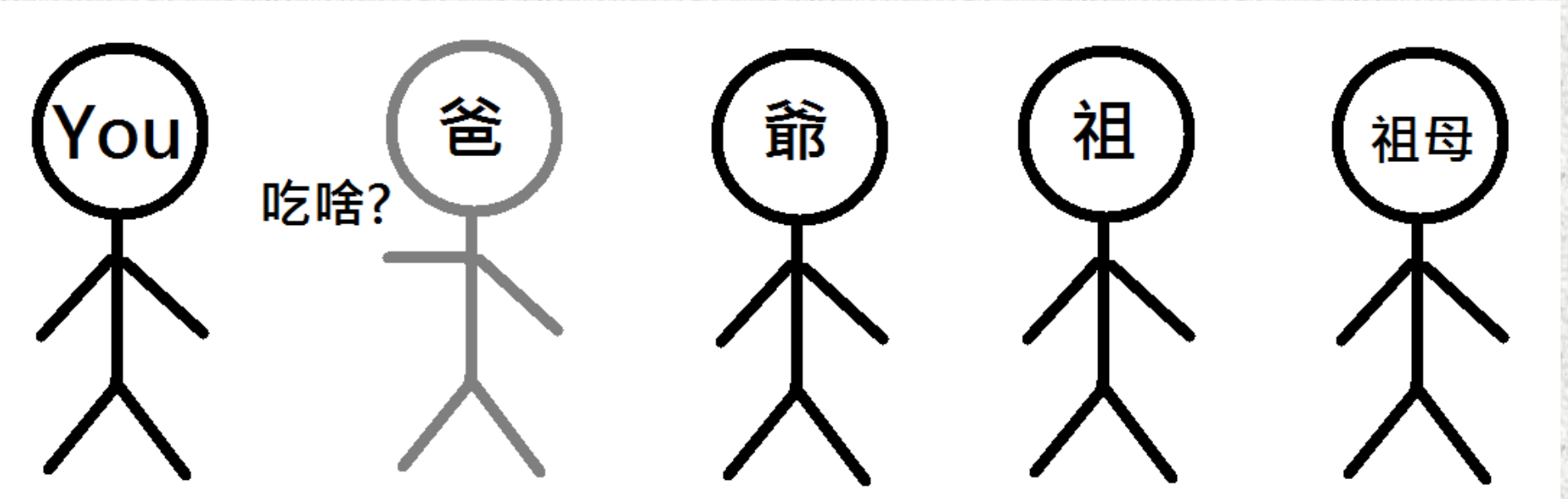
- 然後你祖父就會去問爺爺.....



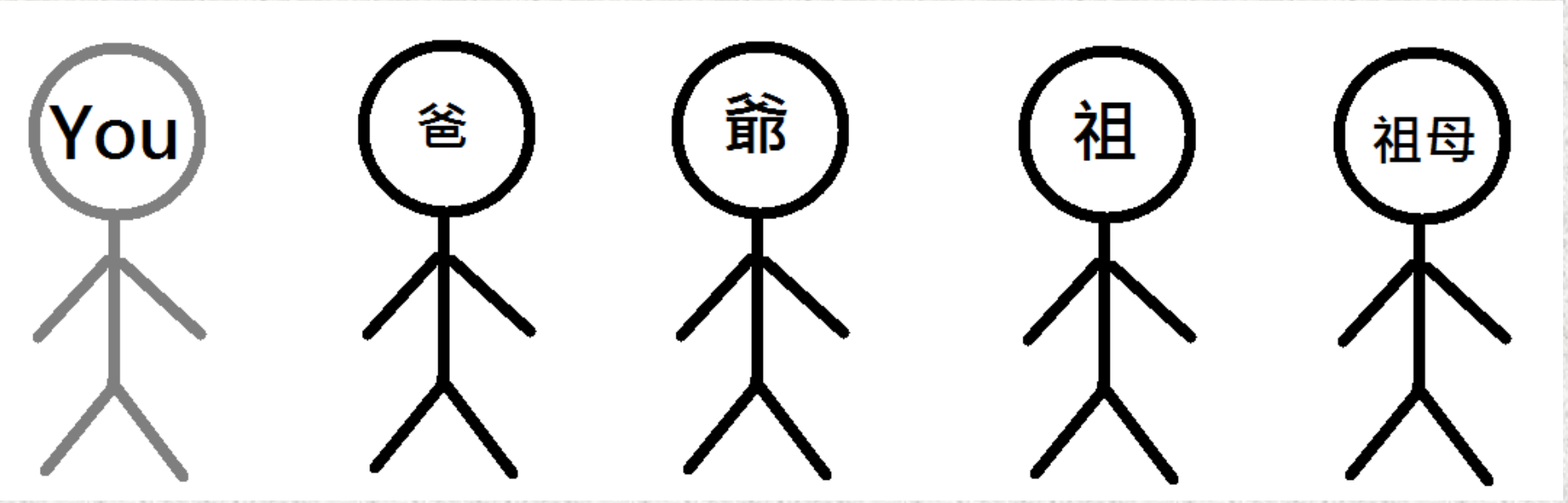
- 然後你爺爺就會去問爸爸.....



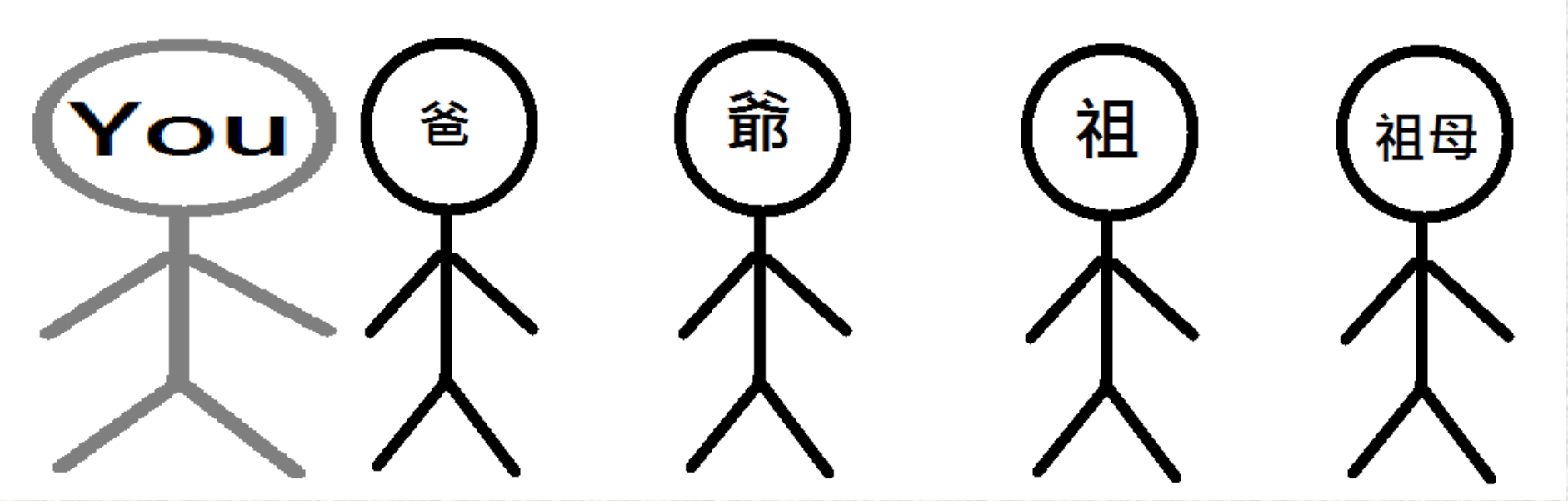
- 然後你爸爸就會去問你.....



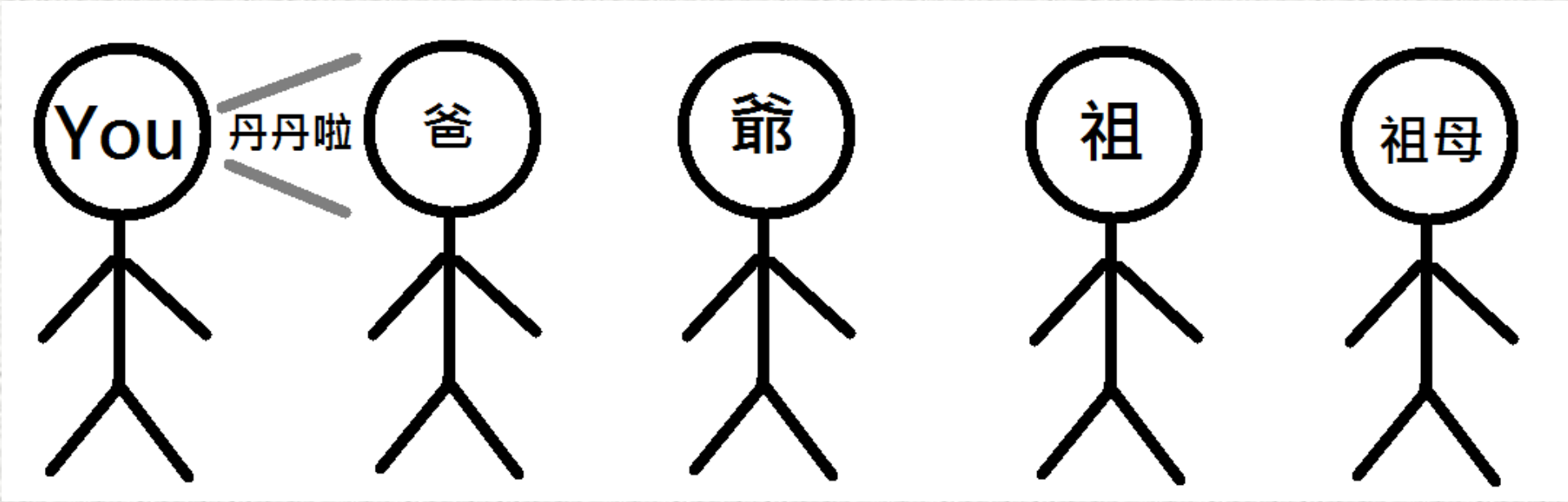
- 然後你就去問.....



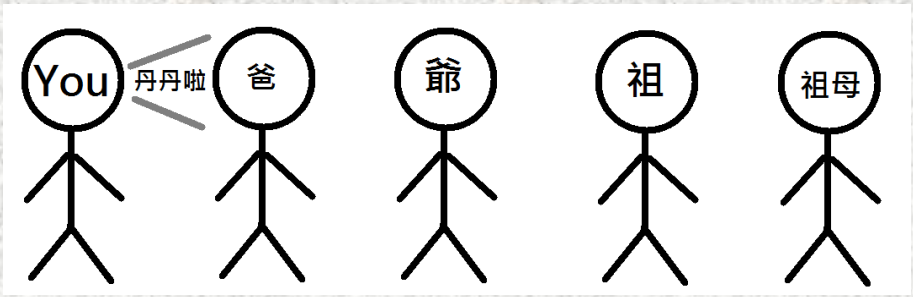
- 沒得問啦!

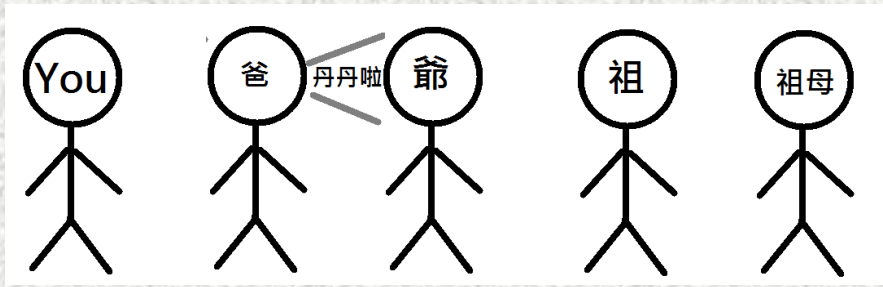


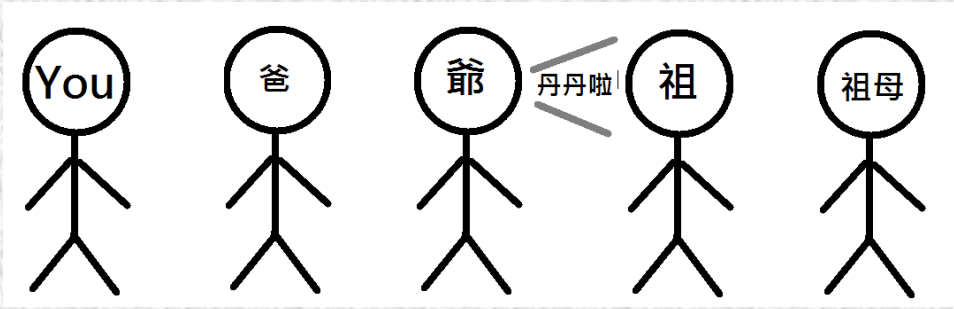
- 你只好回答去吃「丹丹漢堡」

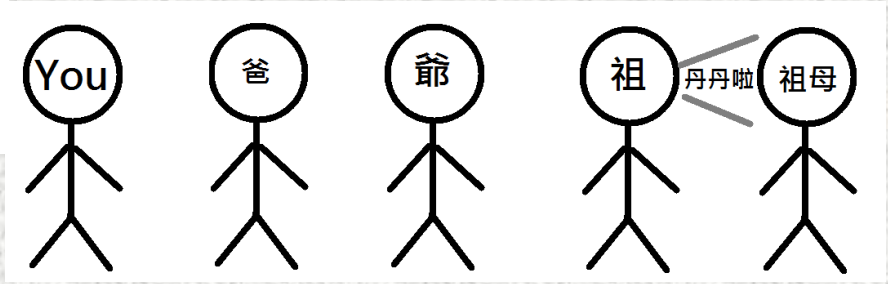


- 然後他們會一直往上回答

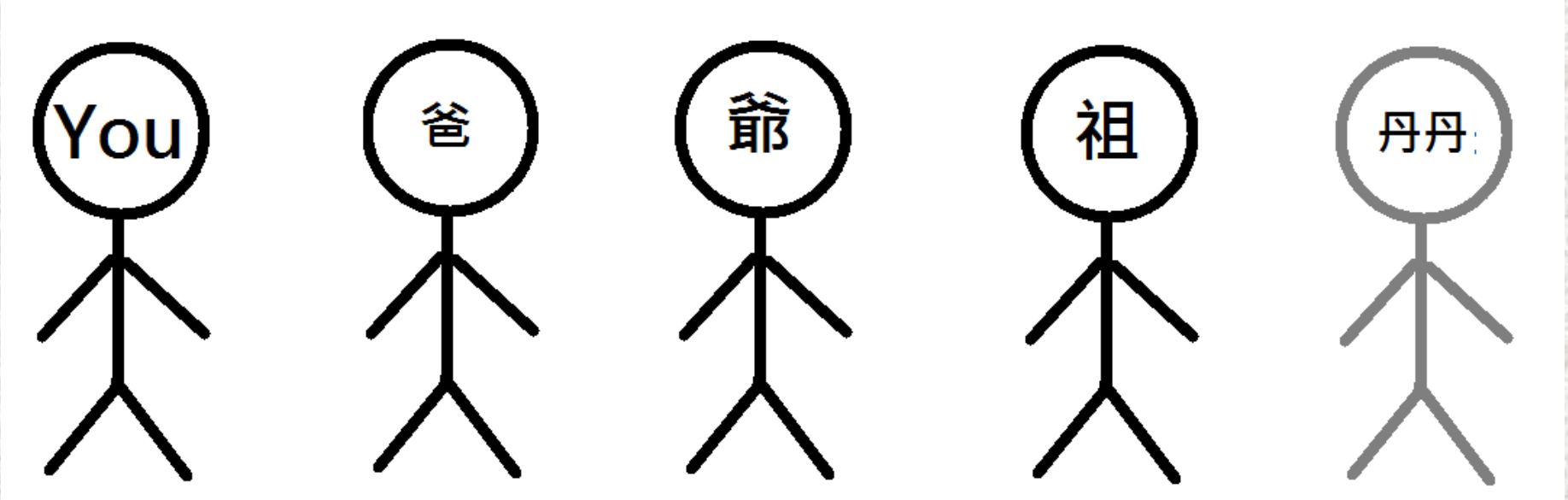






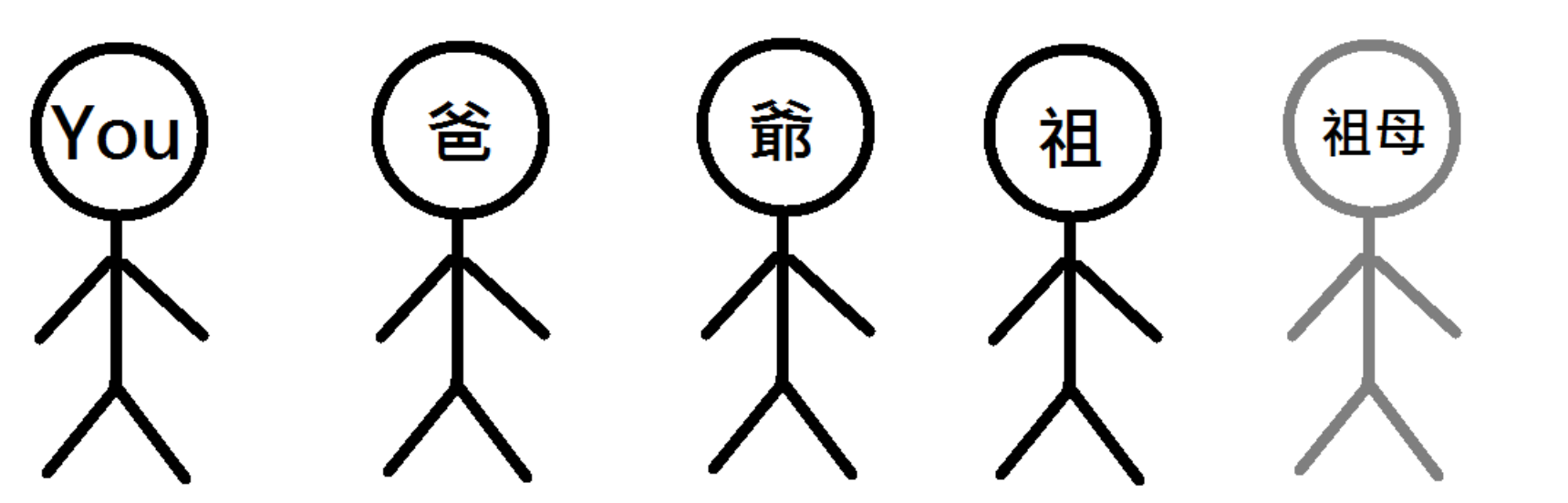


- 最後祖母就知道要吃丹丹漢堡了



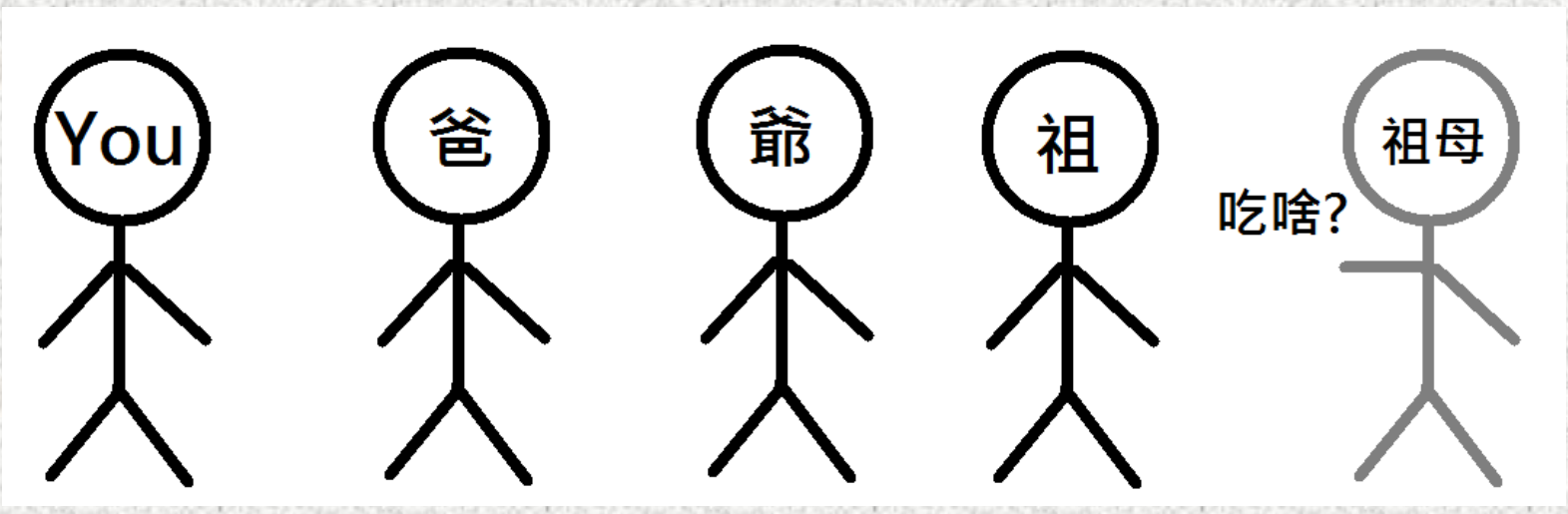
每個人如何回答問題

- 每個人回答問題的方式：
  先問自己的兒子，
  然後把他的答案告訴問的人
  如果沒有兒子就回答丹丹漢堡。



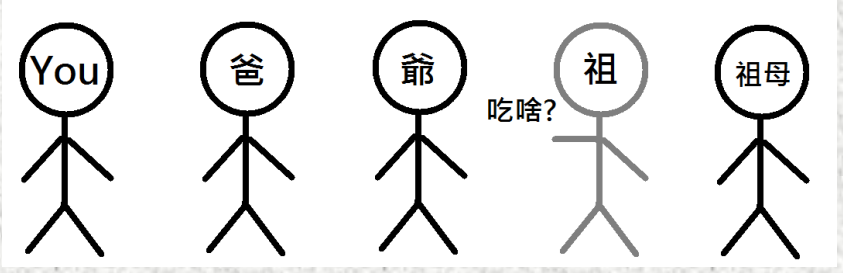
寫成程式吧

```
食物   問要吃什麼(人) {
  if( 我沒有兒子)
        return 丹丹漢堡
  else
        return  問要吃什麼(自己的兒子)
}
```

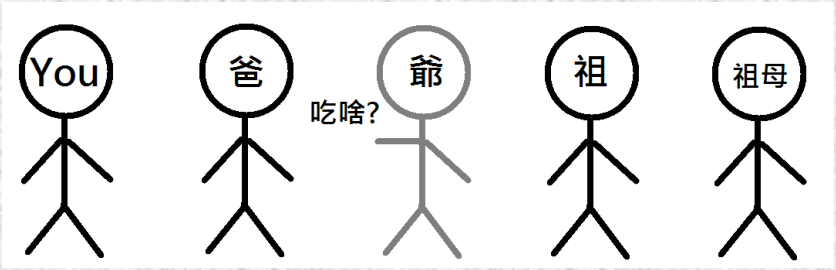


晚上要吃的東西 = 問要吃什麼(祖父)

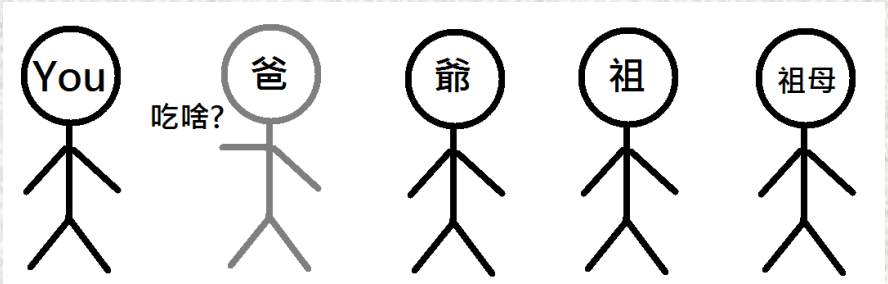
問要吃什麼(祖父)
return 問要吃什麼(爺爺)



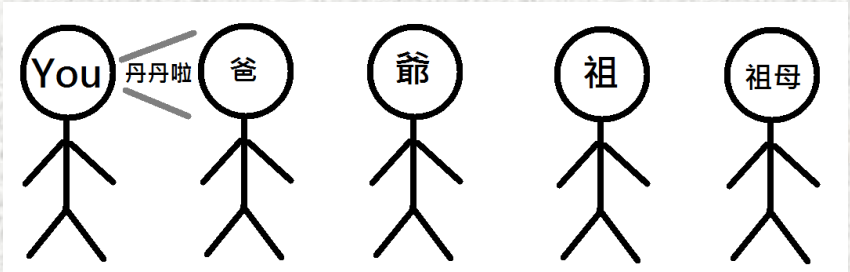
問要吃什麼(爺爺)
return 問要吃什麼(爸爸)



問要吃什麼(爸爸)
return 問要吃什麼(You)



問要吃什麼(You)
return 丹丹漢堡



遞迴的條件

- １.遞迴條件
  要呼叫自己，這樣才有遞迴到，不然就只是普通的函數罷了。
  Ex.問要吃什麼(自己的兒子)
- 2. 終端條件(終止條件)
  因為程式會不停的呼叫自己，如果甚麼都不做的話它永遠也不會停止(或發生錯誤)，所以我們要給一個停下來的條件。
  Ex. if(我沒有兒子)
  return 丹丹漢堡

程式舉例：又是gcd

- 假設 a > b
  - gcd(a,b)
    = b, 如果 b 整除 a
    = gcd(b, a % b), 其他狀況

- 用輾轉相除法就可以解釋！
  當然你不懂也沒差，照樣子coding就好

程式舉例：又是gcd

```
1 int gcd(int a, int b)
2 {
3    if(a % b == 0)                  // 終端條件
4        return b;
5    else
6        return gcd(b, a % b); // 遞迴條件
7 }
8
```

費波那契數列(費氏數列)

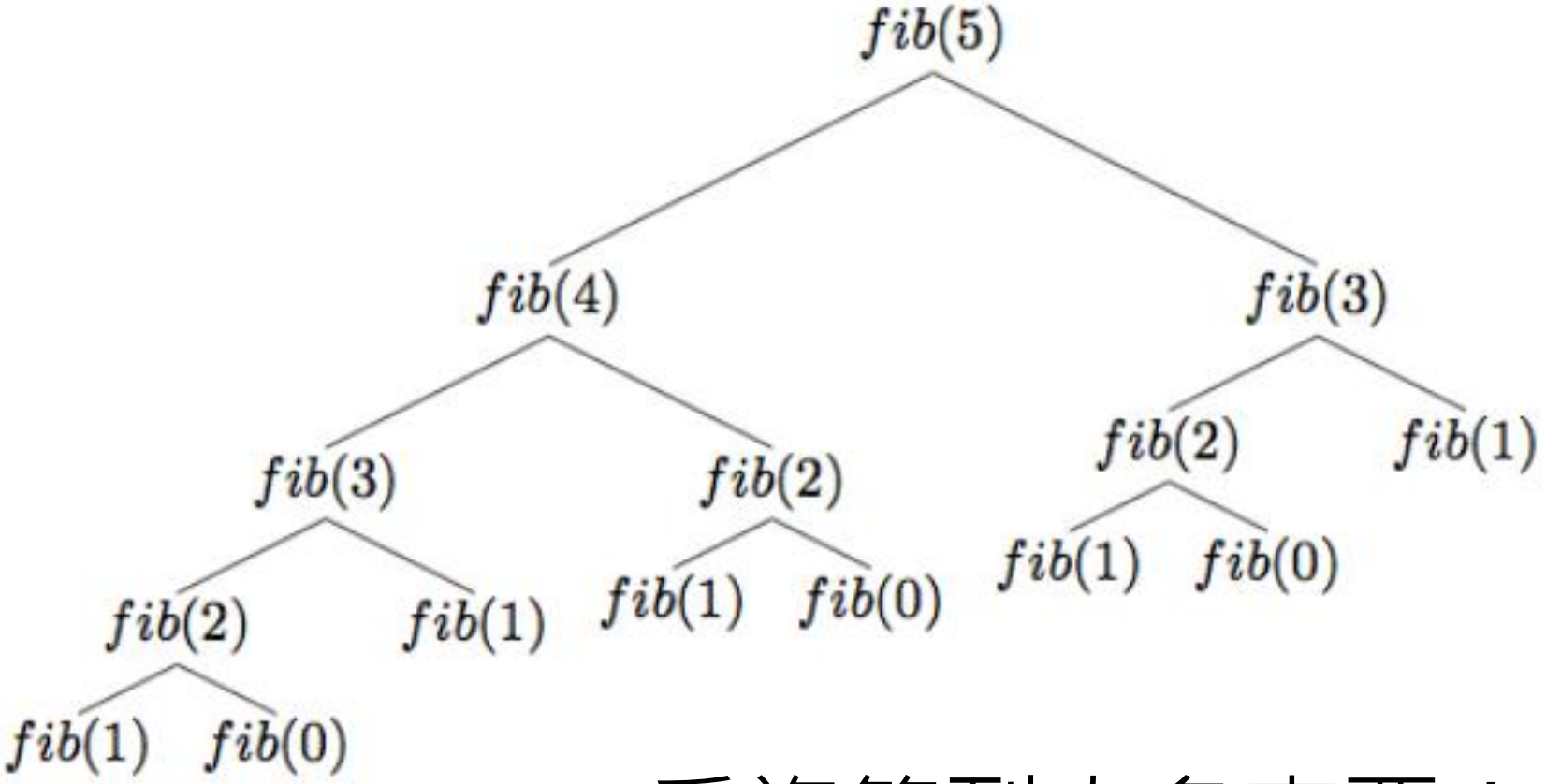
0, 1, 1, 2, 3, 5, 8, 13, 21

終端條件？
遞迴條件？

- $F_0 = 0$
- $F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$（n≧2）

Your Turn
#TOJ 33

費氏數列 – 效率問題



重複算到太多東西！

費氏數列 – 效率問題

- 1. 開個陣列把它存起來
- 2. 使用迴圈做！

總結

- 遞迴函數需要終端條件與遞迴條件！
- 遞迴可以讓程式與數學定義契合，程式碼較為簡潔易懂。
- 遞迴可能會算到許東重複的東西，可以開個陣列存下來節省時間。
- 遞迴的效率通常不高，迴圈比較快，但是有些時候迴圈會比較好做。