

Array VS Recursive

Data Testing:

From the result of date testing, we find that:

1.The time consumed by array way is steady and short with the increase of the length of the Fibonacci sequence;

2.The time consumed by recursive way is increasing with the increase of the length of the Fibonacci sequence, especially when the length is more than 30;

3.When the length of the Fibonacci sequence is small enough, recursive way takes less time, but this advantage disappear once the length is a little bigger, for example, more than 20.

comparison.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Array(19): 0s	Recursive(19): 0.001s
Array(20): 0s	Recursive(20): 0.001s
Array(21): 0s	Recursive(21): 0.001s
Array(22): 0s	Recursive(22): 0.002s
Array(23): 0s	Recursive(23): 0.003s
Array(24): 0s	Recursive(24): 0.006s
Array(25): 0s	Recursive(25): 0.011s
Array(26): 0s	Recursive(26): 0.016s
Array(27): 0s	Recursive(27): 0.025s
Array(28): 0s	Recursive(28): 0.039s
Array(29): 0s	Recursive(29): 0.065s
Array(30): 0s	Recursive(30): 0.103s
Array(31): 0s	Recursive(31): 0.16s
Array(32): 0s	Recursive(32): 0.26s
Array(33): 0s	Recursive(33): 0.421s
Array(34): 0s	Recursive(34): 0.698s
Array(35): 0.001s	Recursive(35): 1.085s
Array(36): 0s	Recursive(36): 1.798s
Array(37): 0s	Recursive(37): 2.839s
Array(38): 0s	Recursive(38): 4.576s
Array(39): 0s	Recursive(39): 7.419s
Array(40): 0s	Recursive(40): 12.006s

Theoretical Analysis:

The reason why array way is usually more efficient than recursive way is that:

When we use an array to generate a Fibonacci sequence, we calculate the element one by one, and for each element we just do this once. However, if we use recursive function to implement this, we have to calculate the elements repeatedly, which is far more time-consuming. The time consumed by recursive function increases **exponentially** with the increase of the index n .

The following fitted curve displays the time consumed by recursive function to generate a Fibonacci sequence that has index elements:

