

用递归法求斐波那契数列过程中，当 $n > 2$ 的时候， $n-1$ 和 $n-2$ 都会再次调用函数运算，然后 $n-1$ 如果大于 2 的话其下面的两项又会再一次调用函数运算，也就是说每一项的运算只要是其前面有两项斐波那契数列的话，都会调用函数，一直到 $n=1$ 和 $n=2$ 为止，这样类似于一个二叉树，每个节点都会调用函数运算，一直到末梢（ $n=1$ 和 $n=2$ ）为止，极大浪费了内存和算力。并且是指数级的增长，所以项数越大，运算速度越慢。

当 $n=1$ 或 $n=2$ 时，	运算一次
$N=3$	函数调用 $A(3)=3$ 次
$N=4$	函数调用 $A(4)=5$ 次
$N=5$	函数调用 $A(5)=9$ 次
$N=6$	函数调用 $A(6)=15$ 次
$N=7$	函数调用 $A(7)=25$ 次
$N=8$	函数调用 $A(8)=41$ 次
$N=n$	函数调用 $A(n-2)+A(n-1)+1$ 次

```
第1项: 1
第2项: 1
第3项: 3
第4项: 5
第5项: 9
第6项: 15
第7项: 25
第8项: 41
第9项: 67
第10项: 109
第11项: 177
第12项: 287
第13项: 465
第14项: 753
第15项: 1219
第16项: 1973
第17项: 3193
第18项: 5167
第19项: 8361
第20项: 13529
第21项: 21891
第22项: 35421
第23项: 57313
第24项: 92735
第25项: 150049
第26项: 242785
第27项: 392835
第28项: 635621
第29项: 1028457
第30项: 1664079
第31项: 2692537
第32项: 4356617
第33项: 7049155
第34项: 11405773
第35项: 18454929
第36项: 29860703
第37项: 48315633
第38项: 78176337
第39项: 126491971
第40项: 204668309
第41项: 331160281
第42项: 535828591
第43项: 866988873
第44项: 1402817465
第45项: 2269806339
第46项: 3672623805
```