**7205 HW6**

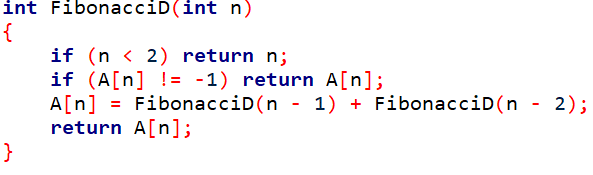
**Name: Xuebao Zhao NUID: 002108354**

**Q1:**

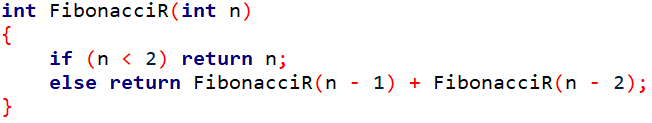
In this problem, I used the main function which was already given. Wrote two approaches: FibonacciR and FibonacciD.

FibonacciR solves the problem recursively where it calls itself recursively exactly twice. FibonacciD solves the problem recursively but with the support of dynamic programming.

FibonacciD:

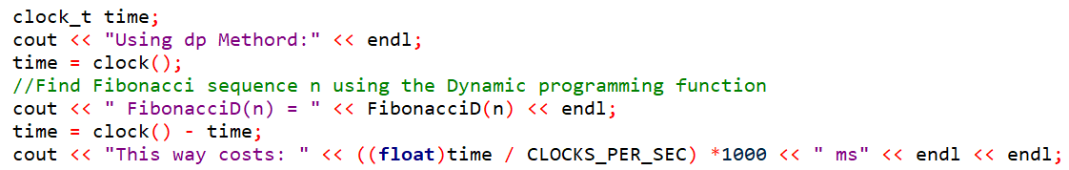


FibonacciR:

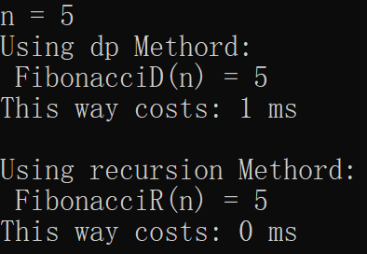
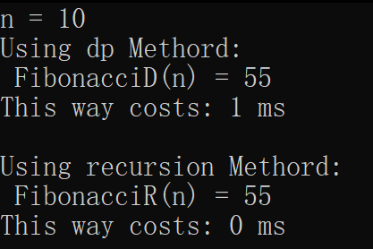
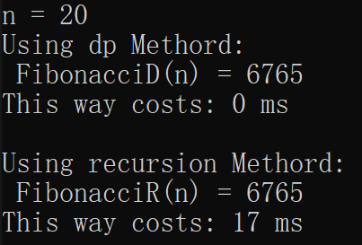


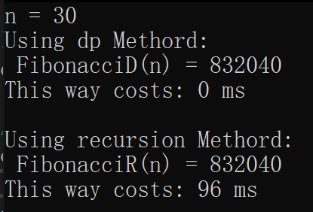
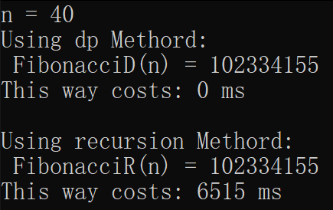
1. **Comment on the running time of each function.**

Using time\_start, time\_end of type DWORD to record the time. And using the time\_start, time\_end to calculate the running time.



Test the code with n values 5, 10, 20, 30, 40:

The results show that when n is less than 20, there is no big difference between recursive method and dynamic programming method. But when n is not smaller than 20, the speed of recursive method drops significantly because recursive method needs to calculate the same value multiple times.

1. **For each function, find the big O asymptotic notation of its running time growth rate.**

**文本, 信件

描述已自动生成** **图片包含 图表

描述已自动生成**

**Q2:**

Recursive method:

**Text

Description automatically generated**

Dynamic Programming method:

**Text

Description automatically generated**

Store price:

**Text

Description automatically generated with medium confidence**

Result:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rod Size | Recursive Time | Recursive Max Revenue | Dynamic Time | Dynamic Max Revenue |
| 5 | 0 | 12 | 0 | 12 |
| 10 | 0 | 25 | 0 | 25 |
| 15 | 0 | 37 | 0 | 37 |
| 20 | 10000 | 50 | 0 | 50 |
| 25 | 410000 | 62 | 0 | 62 |
| 30 | 1.307e+07 | 75 | 0 | 75 |
| 35 | No solution | No solution | 0 | 87 |
| 40 | No solution | No solution | 0 | 100 |
| 45 | No solution | No solution | 0 | 112 |
| 50 | No solution | No solution | 0 | 125 |

Form the table we can see that the speed of dynamic programming is always zero while the speed of recursive method is increasing rapidly.

**Q3:**

**图片包含 图示

描述已自动生成**

**一些文字和图片的手机截图

中度可信度描述已自动生成**

白板上的文字

中度可信度描述已自动生成