

Jared Parkinson
Mod 4
OSU CS162

Non recursive vs the Recursive Fib numbers:

Going into this I assumed that the recursive functions were going to end up being longer of course. I didn't have a time frame to put my expectations to, but I did think it would continue to get a larger gap between the two as time went on.

Below we can see that the Non Recursive time barely jumped up until #22, then it went back down. In contrast, the Recursive started to jump up quite a bit around #27 and steadily rose. **I would not recommend anything past $N = 48$** for testing purposes. In Flip, even though I am using a double it doesn't seem to post the .001 time for Non Recursive. That being said, flip was MUCH faster than my own machine using Visual Studio. I have included visual studio screenshots to show you the output for the Non Recursive.

```
1
-----
14th Fibonacci NonRec #:1
Fib Non Rec Time: 0.001
14th Fibonacci Rec #:1
Fib Rec Time: 0.001
-----
2
-----
24th Fibonacci NonRec #:1
Fib Non Rec Time: 0.001
24th Fibonacci Rec #:1
Fib Rec Time: 0
-----
3
-----
34th Fibonacci NonRec #:2
Fib Non Rec Time: 0.001
34th Fibonacci Rec #:2
Fib Rec Time: 0.001
-----
4
-----
44th Fibonacci NonRec #:3
Fib Non Rec Time: 0.001
44th Fibonacci Rec #:3
Fib Rec Time: 0.001
-----
5
-----
54th Fibonacci NonRec #:5
Fib Non Rec Time: 0.001
54th Fibonacci Rec #:5
Fib Rec Time: 0.001
-----
6
-----
64th Fibonacci NonRec #:8
Fib Non Rec Time: 0.001
64th Fibonacci Rec #:8
Fib Rec Time: 0.001
-----
7
-----
74th Fibonacci NonRec #:13
Fib Non Rec Time: 0.001
74th Fibonacci Rec #:13
Fib Rec Time: 0.001
-----
8
-----
84th Fibonacci NonRec #:21
Fib Non Rec Time: 0.001
84th Fibonacci Rec #:21
Fib Rec Time: 0.001
-----
9
-----
94th Fibonacci NonRec #:34
Fib Non Rec Time: 0.001
94th Fibonacci Rec #:34
Fib Rec Time: 0.001
-----
10
-----
104th Fibonacci NonRec #:55
Fib Non Rec Time: 0
104th Fibonacci Rec #:55
Fib Rec Time: 0.001
-----
```

```
21
-----
214th Fibonacci NonRec #:10946
Fib Non Rec Time: 0.001
214th Fibonacci Rec #:10946
Fib Rec Time: 0.002
-----
22
-----
224th Fibonacci NonRec #:17711
Fib Non Rec Time: 0.002
224th Fibonacci Rec #:17711
Fib Rec Time: 0.003
-----
23
-----
234th Fibonacci NonRec #:28657
Fib Non Rec Time: 0.001
234th Fibonacci Rec #:28657
Fib Rec Time: 0.003
-----
24
-----
244th Fibonacci NonRec #:46368
Fib Non Rec Time: 0.002
244th Fibonacci Rec #:46368
Fib Rec Time: 0.005
-----
25
-----
254th Fibonacci NonRec #:75025
Fib Non Rec Time: 0.001
254th Fibonacci Rec #:75025
Fib Rec Time: 0.008
-----
26
-----
264th Fibonacci NonRec #:121393
Fib Non Rec Time: 0.001
264th Fibonacci Rec #:121393
Fib Rec Time: 0.01
-----
27
-----
274th Fibonacci NonRec #:196418
Fib Non Rec Time: 0.001
274th Fibonacci Rec #:196418
Fib Rec Time: 0.017
-----
28
-----
284th Fibonacci NonRec #:317811
Fib Non Rec Time: 0.001
284th Fibonacci Rec #:317811
Fib Rec Time: 0.025
-----
29
-----
294th Fibonacci NonRec #:514229
Fib Non Rec Time: 0.001
294th Fibonacci Rec #:514229
Fib Rec Time: 0.038
-----
30
-----
304th Fibonacci NonRec #:832040
Fib Non Rec Time: 0.001
304th Fibonacci Rec #:832040
Fib Rec Time: 0.063
-----
```

```
33
-----
334th Fibonacci NonRec #:3524578
Fib Non Rec Time: 0.001
334th Fibonacci Rec #:3524578
Fib Rec Time: 0.259
-----
34
-----
344th Fibonacci NonRec #:5702887
Fib Non Rec Time: 0.001
344th Fibonacci Rec #:5702887
Fib Rec Time: 0.421
-----
35
-----
354th Fibonacci NonRec #:9227465
Fib Non Rec Time: 0.001
354th Fibonacci Rec #:9227465
Fib Rec Time: 0.676
-----
36
-----
364th Fibonacci NonRec #:14930352
Fib Non Rec Time: 0.001
364th Fibonacci Rec #:14930352
Fib Rec Time: 1.097
-----
37
-----
374th Fibonacci NonRec #:24157817
Fib Non Rec Time: 0.002
374th Fibonacci Rec #:24157817
Fib Rec Time: 1.773
-----
38
-----
384th Fibonacci NonRec #:39088169
Fib Non Rec Time: 0.001
384th Fibonacci Rec #:39088169
Fib Rec Time: 2.886
-----
39
-----
394th Fibonacci NonRec #:63245986
Fib Non Rec Time: 0.001
394th Fibonacci Rec #:63245986
Fib Rec Time: 4.714
-----
```

By 38 and 39 it starts to explode and almost double. If you accidentally choose N much higher than 50 I would definitely hit Ctrl+Z to suspend the program.

Factorials

For factorials, you will want to use something between **500,000,000** and **1,000,000,000** to see any form of results. If you use greater than 1 billion, it will break my menu system. At first I was using 40 like the Fib numbers but I didn't understand you needed to use a very large number to make this show anything.

```
flip1 ~/CS162/10week/mod4 201% mod4
Enter desired N:
1000000000

-----MAIN MENU-----
Choose from the following options:
1. Fib Numbers
2. Factorials
Enter Choice: 2

-----
FACTORIALS
-----
rfactorial Time: 1.89 Num: 0
factorial Time: 1.88 Num: 0
flip1 ~/CS162/10week/mod4 202% █
```

With the factorials, it looks pretty much like I expected. If you use 500,000,000 it will show almost no difference. As it gets closer to 1,000,000,000 it will start to increase. I would assume that the only way to see significant differences in this is to have a number so large that you will forget what digit you are on when typing it.

All-in-all, this turned out as expected. I knew that the recursive functions would take longer, but honestly if you are using the factorial stuff in real life, you wouldn't need to worry about it unless you were using a massive number of recursions. On the Fib numbers however, this shows up VERY soon and would be a horrible efficiency level. Your software would be turtle slow and people would not want to use it.