Shashank K

SE21UCSE198

CSE 3

DAA Assignment -2

Goal : Comparing the time taken by a recursive function and a non-recursive function for finding the value of function 2^n.

Non recursive method

Function :

```
def multiply(n):
  answer = 1
  while n:
    answer*=2
    n-=1
  return answer
```

The time taken for various values of n is as follows.

Value of n	Time taken in seconds
1	2.15e-06
10	2.62e-06
100	1.71e-05
500	8.08e-05
955	1.82e-04

Recursive Method

Function :

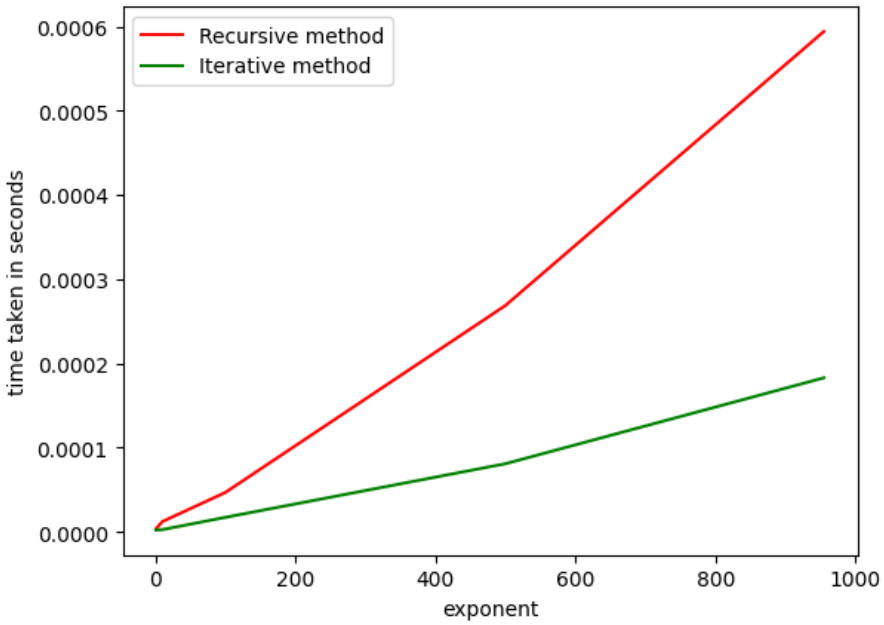
```
def exponent(n):
    if n == 0:
        return 1
    else:
        return 2 * exponent(n - 1)
```

The time taken for various values of n is as follows.

Value of n	Time taken in seconds
1	3.57e-06
10	1.21e-05
100	4.67e-05
500	2.61e-04
955	5.41e-04

Observations:

- I was not able to test for higher values of n because of the recursion limitation in python. But it would've been completely possible with looping method.
- Just from the table we can clearly see that in every case, looping method has proven to be faster than recursive method and at higher value of n it has proven to be even better.



- The reason for this might be the depth of recursion, repeated number of function calls. Each function occupies a lot of memory hence the poor performance.

Calculating Complexity :