Problem 5.1

d)

Fibonacci Sequence
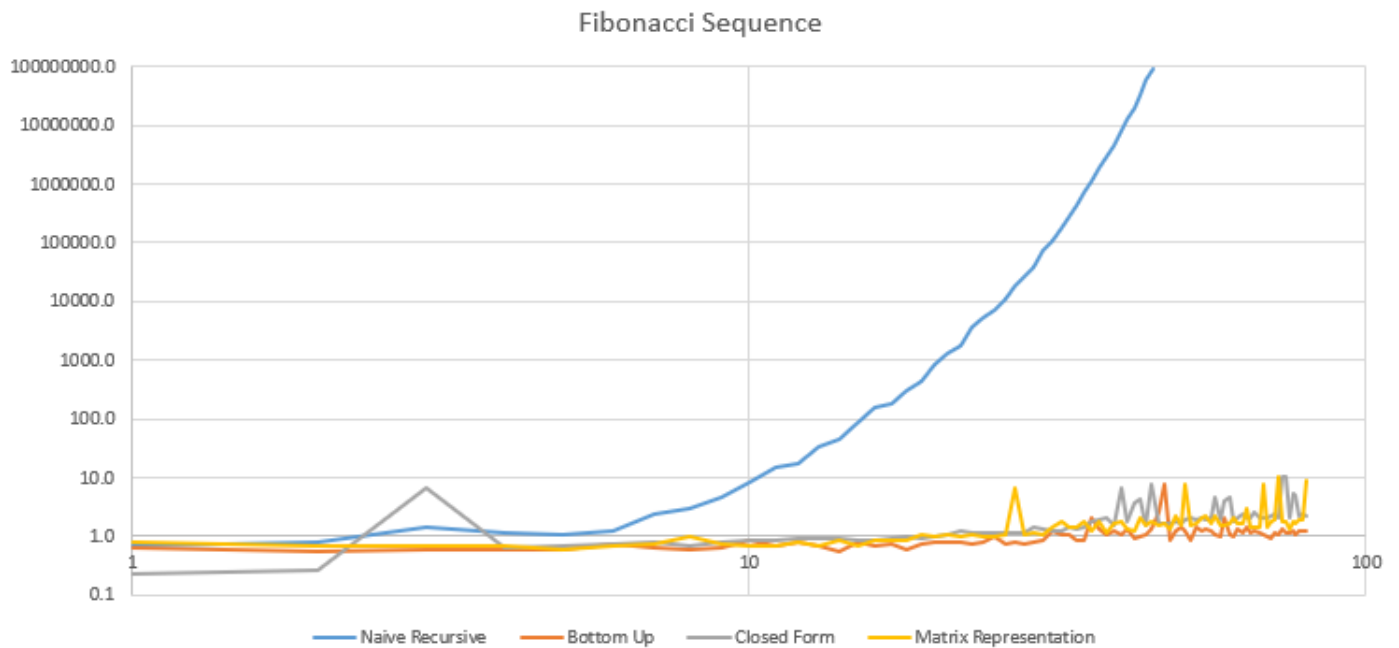

Among these 4 methods of getting n'th Fibonacci value,

'Naïve Recursive' algorithm bounds to O(n^2), which is obviously time consuming then other three methods.

Except the one, other three methods (Bottom Up, Closed Form, Matrix Representation) seems to be bounded by O(lg n) - or O(n) -

which is much more efficient on time-complexity comparison.

Problem 5.2

d)

$$T(n) = 4(n/2) + f(n)$$

0

$h = \log_2 n$

$T(n)$

$a = 4$

1

$T(\frac{n}{2})\ T(\frac{n}{2})\ T(\frac{n}{2})\ T(\frac{n}{2}) \rightarrow b = 2$

2

$T(\frac{n}{2^2})$ ''

$\vdots$

h

$T(1)$

$\cdots \rightarrow a^h \cdot T(1) = n^{\log_b a} \cdot T(1)$ = O(n²)