Assignment 3

Jovan Shandro

Problem 3.1

a) b) can be checked in the folder named “Problem 1

c)For same values of n all the methods return the same number except for the closed method which for large values of n gives a different value.This happens because the representation of the fibonacci numbers as powers of rationals is slightly different and because of that we have to round the value, but for relatively large n we can notice that they do not give the same number as rounding is not powerful enough. So the change is because of the rounding

d)Comparing the three results we can clearly see how exponentially slower

naive recursive method is compared to the others. The other methods perform relatively fast and are pretty similar in their computational times. Overall the closed method and the matrix are slightly faster than the bottom up method .

Problem 3.2

a) The brute force approach is the multiplication algorithm we all learn at school in which we multiply the digits (bits in our case) of the second number from left to right with the first number and add them in the end considering the corresponding shifts due to multiplication by 10 (in our case 2). Since we have to iterate to every bit of the second number and for each of them iterate througth all the bits of the first one we end up with a time complexity of O(n^2). An implementation of the algorithm can be found on the folder “Problem 2”.

b) Using a divide and conquer approach, we split both the numbers into 2 halves thus since it is given assuming n is even and letting A B be the 2 numbers we have $A = 2^{n/2} * A\_left + A\_right$ and $B = 2^{n/2} * B\_left + B\_right$ where A_left and A_right contain the n/2 leftmost and rightmost bits (same for B respectively). So we get

$A*B = 2^{n} A\_left*B\_left + 2^{n/2} * [(A\_left + A\_right)(B\_left + B\_right) – A\_left*B\_left – A\_right*B\_right] + A\_right*B\_right$

Please check the implementation done in c++

c)Since as we see in both implementation and above the recursive formula is

$T(n) = 3T(n/2) + \theta(n)$ as the method calls itself 3 times recursively with half of the initial values and the conquering method is realised in $\theta(n)$ as it is given than multiplication and addition between these numbers in bits is realised in $\theta(n)$.

d)e) check the pdf in paper as the proofs were more understandable written in hand.