CS566 HW2 Jixing Jacey Man

Question 1:

C = (1 3   (6 8
7 5) *   4 2)

C = (1*6 + 3*4   1*8+ 3*2
7*6 + 5*4   7* 8 + 5*2)

C = (6 + 12 8+6
42 + 20   56 + 10)

C = (18 14
62 66)

Question 2:

Pesudocode for Strassen:

SquareMatrixMultipy(A,B)
N = A.rows
Let C be a new n * n matrix
If n == 1 then
    C11 = A11*B11
Else Partition A,B, and C
#getting sums
sum1= A11 + A12
sum2 = A21 + A22
sum3 = B12- B22
sum4= A11 + A22
sum5 = B21 – B11
sum6 = A12 – A22
sum7 = B11 + B22
sum8 = A11 – A21
sum9 = B21 + B22
sum10 = B11 + B12

#getting products
p1 = SquareMatrixMultipy(A11,sum3)
p2 = SquareMatrixMultipy(sum3,B22)
p3 = SquareMatrixMultipy(sum2,B11)
p4 = SquareMatrixMultipy(A22,sum5)
p5 = SquareMatrixMultipy(sum4,sum7)
p6 = SquareMatrixMultipy(sum6,sum9)
p7 = SquareMatrixMultipy(sum8,sum10)

C11 = p4 + p5 – p2 + p6
C12 = p1 + p2
C21 = p3 + p4

C22 = p1 – p7+p5 – p3

end if
    return C
#EndFunction

Question 3:

Answer: Yes, it is possible to use Strassen's algorithm to compute this following 3*3 matrix.

Is it possible to use Strassen's algorithm for any matrix multiplication?
Answer: Yes

Computation Step: I would first get products using cross multiplication, than get the sum of the products to get the final C(3X3) matrix.

1. So divided the input matrices A and B and output matric C into n/3 * n/3, this takes Theta(1)
2. Create the Sum and Product equation to get all the individual number.
   By cross – multiplying A and B there should be 9 X 9 from A and B, total 18 number from Products.
3. Then adding all 18 number into 9 numbers matric would results the final C.

Is it possible to use Strassen's Algorith for any matrix multiplication?
Answer: Yes

What is the resulting algorithm run time in Theta notation?
Answer: The running Theta time for 3X 3 would be Theta(n^3) time, so for any matrix multiplication would be Theta(n^n) for running time.

Question 4:

1. Show that the solution of $T(n)=T(n-1)+n$ is $O(n^2)$
   Solution: solution guess is : T(n) <= cn^2
   Assuming c and n0 are positive constants > 0, and prove T(n) <= cn^2
   T(n) <= c(n – 1)^2 + n
   = cn^2 – 2cn + c + n
   = cn^2 – c(1-2c)+c
   <= cn^2 holds as long as c >= 1

2. Show that the solution of

T(n)=T([n/2])+1 is O(lgn).

Solution: solution guess is T(n) <= c*lg(n-2)
Assuming c and n0 are positive constants > 0, and prove T(n) <= c*lg(n-2)
T(n) <= clg([n/2] – 2) + 1
<= clg(n/2 + 1-2) + 1
= clg((n-2)/2)+1
=clg(n-2)-clg2+1
<=clg(n-2) holds as long as c>= 1

Question 5:
If a = b^d is case 1
If a < b^d is case 2
If a > b^d is case 3
For all the recurrences below, a = 2, and b = 4, therefore, n^(log base^a) = sqrt(n)

1. T(n) = 2T(n/4) + 1
   F(n) = O1 = O(n^(0.5-0.5) = case 1
   Answer: Theta(n^(log base 4^2)) = T(n) = Theta(n^0.5)

2. T(n) = 2T(n/4) + sqrt(n)
   F(n) = O(n^0.5) = case 2
   Answer: Theta(n^(log base 4^2) * log n) = T(n) = Theta(sqr(n^0.5*log n)

3. T(n) = 2T(n/4) + n^2
   F(n) = O(n^2) = O(n^(0.5+1.5)) = case 3
   Answer: T(n) = Theta(n^2)