Written Questions:

1) (15 points) For the following snippets of code, compute the running time function T(n) by determining the number of times each line executes and computing (no O-notation). When relevant, determine the best case and the worse case running time.

(a) Cost. Times

```
//Input: A nonnegative integer n
S = 0
for i = 1 to n do
      S = S + i * i
return S
```

(b)
```
//Input: An array A[0..n-1] of n real numbers
minval = A[0]
maximal = A[0]
for i = 1 to n-1 do
      if A[i] < minval
            minval = A[i]
      if A[i] > maxval
            maxval = A[i]
return maxval - minval
```

(c)
```
//Input: A matrix A[0..n-1, 0..n-1] of real numbers
for i = 0 to n-1 do
    for j = i+1 to n-1 do
          if A[i, j] ≠ A[j, i]
                return false
return true
```

Programming Questions:

2) (40 points) Insertion Sort and merge sort implementation:

(a) Implement the insertion sort algorithm described in the book and in class

(b) Test your code by generating arrays of random numbers of increasing size (you can do this easily with a loop) and verify that running time is cn^2. You can verify this by dividing the runtime values you get by n^2 and the result would be a constant. For example if for an array of size 1000 your algorithm takes x ms, you divide x by 1000^2

(c) Test your code by creating arrays of sorted values (1, 2, .., n for example) and verify that the algorithm runs in cn

(d) Implement the merge-sort algorithm described in class.

(e) Test your code by using arrays of randomly generated values and verify runtime is $cnlgn$

A)
C1 : 1
C2 : N+1
C3 : N
C4: 1
Total:
C1 * 1 + C2 * (N+1) + C3 * (N) + C4 * 1
= C1 + C2 N + C2 + C3 N + C4
= C1 + (C2 + C3) N + C2 + C4
= (C2 + C3) N + (C1 + C2 + C4)
T(a)= a n + b

B)
C1 : 1
C2 : 1
C3: N
C4 : N-1
C5 : tj
C6 : N-1
C7 : tj
C8 : 1
Total:
C1 * 1 + C2 * 1 + C3 * (N) + C4 * (N-1) + C5 * (tj) +
C6 * (N-1) + C7 * (tj) + C8 * 1
Best case scenario:
C1 * 1 + C2 * 1 + C3 * (N) + C4 * (N-1) + C6 * (N-1) + C8 * 1
C1 + C2 + C3 N + C4 N – C4 + C6 N – C6 + C8
(C3 + C4+ C6) N + (C1 + C2 + C8) – (C4 +C6)
T(a)= a N + b – d
Worst case scenario:

C1 * 1 + C2 * 1 + C3 * (N) + C4 * (N-1) + C5/2 * (N-1) + C6 * (N-1) + C7/2 * (N-1) + C8 * 1
C1 + C2 + C3 N + C4 N – C4 + C5/2 N – C5/2 + C6 N – C6 + C7/2 N – C7/2 + C8
= (C3 + C4+ C5/2+ C6 + C7/2) N + (C1 + C2 + C8) – (C4+ C5/2 +C6 + C7/2)
T(a)= - a n + b – d

C)
C1 : N+1
C2 : SUM j=1 to n-1. Formula j
C3: SUM j=1 to n-1. Formula j-1
C4: SUM j=1 to n-1. Formula tj
C5: 1
Formulas:
C2:-
SUM j=1 to n-1. Formula j
n-1 + 1 to n-1
=n - i or n-n?
C3:-
SUM j=1 to n-1. Formula j-1
n-1 to n-1
=n – (i + 1) or n-n?
Total:
C1 * (N+1) + C2 * (SUM j=1 to n-1. Formula j) + C3 * (SUM j=1 to n-1. Formula j-1) + C4 * (SUM j=1 to n-1. Formula tj) + C5 * (1)

Best case scenario:
C1 * (N+1) + C2 * ((N+2)(N-1)/2)+ C3 * (N(n-1)/2)) + C5 * (1)
C1 * (N+1) + C2 * (N^2-N + 2N -2/2)+ C3 * (N^2-N/2)) + C5 * (1)
C1 N + C1 + C2/2 N^2 + C2/2 N -C2 + C3/2 N^2 - C3/2 N + C5
(C2/2 + C3/2) N^2 + (C1 + C2/2 – C3/2) N + (C1 -C2 + C5)

$T(a)= a N^2 + b N + (C1 - C2 + C5)$

Worst case scenario:

$C1 * (N+1) + C2 * ((N+2)(N-1)/2)* n + C3 * (N(n-1)/2) * n) + C4 *(N-2)(N-1)/2 + C5 * (1)$

$C1 * (N+1) + C2 * (N^2-N + 2N -2/2)* n + C3 * (N^2-N/2) * n) +C4 * (N^2 -N -2N +2/2) + C5 * (1)$

$C1 N + C1 + C2/2 N^2 + C2/2 N + C3/2 N^2 - C3/2 N + C4/2 N^2 -3/2 C4 N + C4 + C5$

$(C2/2 + C3/2 + C4/2) N^2 + (C1- C2/2 - C3/2 – 3/2 C4) N + (C1+C4+C5)$

$T(a)= a N^2 + b N + (C1+C4+C5)$