Design and Analysis of Algorithms

Homework # 1

Total Marks = 65

Q1) Suppose we are comparing implementations of insertion sort and merge sort on the same machine. For inputs of size n, insertion sort runs in $4n^2$ steps, while merge sort runs in $32\,n \lg n$ steps. For which values of n does insertion sort beat merge sort? [5 Marks]

Q2) What is the smallest value of n such that an algorithm whose running time is $100n^2$ runs faster than an algorithm whose running time is 2^n on the same machine? [5 Marks]

Q3) Perform step-count analysis on the following code fragments. Indicate the time taken by each line of code over the life of the program, then add all individual times to get T(n). Where applicable, work in the worst case scenario. Then find an appropriate O(f(n)) for each T(n). In order to do this, you must show must that there exists a positive constant c>0, such that: T(n) <= c f(n). [5*3 = 15 Marks]

(a)
```
int s, i,n;
cin>>n;
s = 0;
for (i=n;i>=1;i--)
    s++;
```

(b)
```
 int sum,i,j,n;
sum = 0;
cin>>n;
for (i=1;i<n;i=i*2)
   for (j=1;j<n;j=j*2)
sum ++
```

c)
```
int x = 0, j = n;
while (j > 0) {
      x += j*3;
      j /= 4;
}
```

Q4) Consider sorting n numbers stored in array A by first finding the smallest element of A and exchanging it with the element in $A[1]$. Then find the second smallest element of A, and exchange it with $A[2]$. Continue in this manner for the first n - 1 elements of A. Write pseudo-code for this algorithm, which is known as ***selection sort***. Why does it need to run for only the first *n* - 1 elements, rather than for all *n* elements? Give the best-case and worst-case running times of selection sort in Θ-notation. [5 Marks]

Q5) Prove that T(n) is Θ (n^3) by finding appropriate constants. [5 Marks]

$$T(n) = \frac{1}{8}n^3 - 5n^2$$

Q6) What is the runtime of the following function? Express your answer using the big-O notation. Show all working [5 Marks]

```
Function Mystery (n)
{
   If (n > 1)
   {
         Print "hello"
         Mystery(n/5)
         For (i=1 …. n)
             Print "world"
         Mystery(2n/5)
    }
}
```

Q7) Use a recursion tree to determine a good asymptotic upper bound on following recurrences. Please see Appendix of your text book for using harmonic and geometric series. (4*5 = 20 Marks)

a) T (n) = 2T(n/4) + O(lgn)

b) T (n) = 3T(n/2) + $O(n)^3$

c) T (n) = 7T(n/5) + Θ (1)

d) T (n) = 2T (n/2) + n/ lg n

e) T (n) = 3T (n - 1) + Θ (1)

Q8) Do a dry run of count inversion pairs algorithm using divide and conquer approach on following array.

A = [5 8 3 9 2 6 4 1 7 0]

Create a recursion tree of the array and write left, right and split inversion pairs each recursive call of the algorithm. [5 Marks]