Design and Analysis of Algorithms

Homework # 1

Total Marks = 95

Q1) 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). [4*5 = 20 Marks]

Part (a)

```
a = n;
while(a > 1) {
        b = 1;
        while (b < n) {
                k = 0;
                while (k < n) {
                        k = k + 2;
                }
                 b = b * 2;
        }
        a = a / 2;
}
```

Part (b)

```
a = 1;
while(a < n/2) {
        b = 1;
        while (b < n) {
                b = b * 2;
        }
        a = a * 2;
}
```

Part (c)

```
a = 1;
while(a < n) {
        b = n;
        while (b > 1) {
                b = b - 3;
        }
        a = a + 4;
}
```

Part (d)

```
a = 0;
for (i=1;i<n;i=i*2)
    for (j=1;j<n;j=j*2)
                a ++
```

Q2) Prove or disprove the following: [3*5 = 15 Marks]

(a)

$$\mathbf{2^{2n} = O\ (2^n)}$$

(b)

$$\mathbf{8^n = O\ (4^n)}$$

(c)

$$\mathbf{3^{n+5} = O\ (3^n)}$$

Q3) Suppose we have a sorting algorithm with following pseudocode: [10+5 = 15 Marks]

If the sequence length is at most 4,
sort it using bubble sort.
Else:
i. Divide the list into 5 pieces evenly, by scanning the entire list.
ii. (recursively) sort the first 3/5 of the list.
iii. (recursively) sort the last 3/5 of the list.
iv. (recursively) sort the first 3/5 of the list.

For example, on the input sequence 1, 5, 3, 2, 4 The first recursive sort produces 1, 3, 5, 2, 4, the second sort produces 1, 3, 2, 4, 5, and the last produces 1, 2, 3, 4, 5.
(a) Write down a runtime recurrence for this sorting algorithm and analyze its asymptotic running time.
(b) Give an example sequence on 5 or 10 integers where this sorting algorithm does not terminate with the correct answer.

Q4) Prove that T(n) is $\Theta\ (n^4)$ by finding appropriate constants. [10 Marks]

$$T(n) = \frac{1}{6}\, n^4 - 4n^2$$

Q5) What is the runtime of the following function? Express your answer using the big-O notation. [5 Marks]

```
Function Mystery (n)
{
   If (n > 1)
   {
        Print "hello"
        Mystery(n/3)
        For (i=1 ….  n)
            Print "world"
        Mystery(n/3)
    }
}
```

Q6) What is the smallest value of n such that an algorithm whose running time is $50n^2$ runs faster than an algorithm whose running time is 2^n on the same machine? [5 Marks]

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. (5 *5 = 25 Marks)

a) $T(n) = T(n/5) + O(n)^2$

b) $T(n) = 10T(n/2) + O(n)^2$

c) $T(n) = 10T(n/2) + \Theta(1)$

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

e) $T(n) = 2T(n - 1) + \Theta(1)$