Eastern University, Sri Lanka

Second Examination in Science (2003/2004)

First Semester

CS 201 – Data Structure & Design of Algorithm

Time allowed: Two hours **Answer All Questions**

Q1.

1. For each of the following complexity classes, name an algorithm treated in this course that belongs in the class

   a. O (1)
   b. O $(\log_2 n)$
   c. O (n)
   d. O $(n \log_2 n)$
   e. O (n^2)
   f. O (2^n)

2. A group of business people have arranged a meeting. As each of them arrives for the meeting, he/she shakes hands with all those already present (*we assume they arrive individually, not simultaneously).* Use induction to show that if **n** people came to the meeting, then $n(n-1)/2$ handshakes occur. (*When two people shake hands, that is counted as one handshake.)*

3. Is $2^n = O(n!)$?. Show your reasoning. (*Recall that* $n! = 1 \times 2 \times 3 \times \ldots\ldots \times n$. *Also that n! = n (n-1)! And that 0! = 1.)*

Q2.

1. Briefly describe the ***MergeSort*** algorithm. (*If you find it difficult to give the details of* ***merge****, you can illustrate its working by an example.*)

2. Explain why the time complexity function **T** for MergeSort is given by the recurrence equations

$$T(1) = 1$$
$$T(n) = 2T(n/2) + n$$

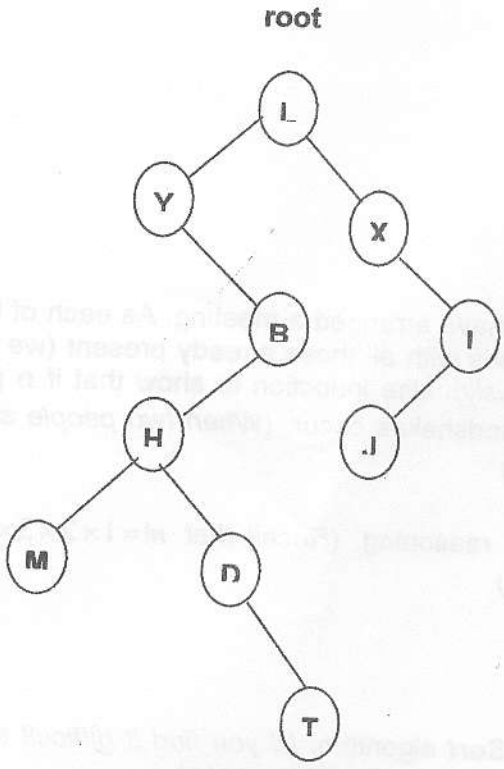
3. Use the iteration method to show that **T** is of $O(n \log n)$ (*It is enough to consider cases* $n = 2^k$ *for k = 0, 1, 2……..)*

Q3.

1. Draw the **final** *binary search tree* **T** that results from successively inserting the letters **'o'**, **'l'**, **'r'**, **'n'**, **'p'**, **'m'** and **'q'** into an initially empty tree.

2. Draw the result of deleting **'o'** from **T**.

3. List the nodes of the following binary tree in ***pre-***, ***in-***, ***post-***, and ***level*** orders



Q4.

1. Explain with the aid of real world problems, the concept of ***Recursion.***

2. Find an ***appropriate name*** for the following methods

a)
```
int f1(int n)
{
        return n * f1(n-1);
}
```

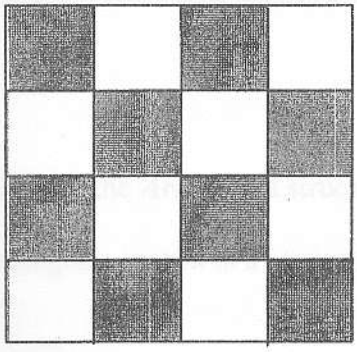
b)
```
int f2(int a, int b)
{
        if (a < 0) return f2(-a,-b);
        else if (a = = 0) return 0;
        else return f2(a-1, b) + b;
}
```

c) *int* ***f3****(int n, int k)*

```
{       //assume k >= 0
        if (k = = 0) return 1;
        else return n * f3(n, k-1);
}
```

2) Describe briefly the ***Backtracking*** technique.

3) Consider the problem of how to place ***4 queens*** on a ***4x4*** chessboard in such a way so that no queen may capture another. *[Recall that the rules of chess do not allow a queen to take another queen that lies on the same row, the same column or the same diagonal (in either direction)].*



4 x 4

Find appropriate solutions to the above problem using the ***STACK*** data structure.