Data Structures and Algorithms – CS2402

Lab 1

due by September 16, 2005, at 5pm
to be sent electronically to the TA

Algorithm Analysis

This first lab will give you the opportunity to play around with **known data structures**, and to become more familiar with **algorithm analysis**.

This lab assignment is composed of two parts. Each part is a separate problem that you have to solve by implementing the corresponding solving method. For both problems, you will have to write down a report (short report, about 3-4 pages per problem) in which you will:

- describe the problem you are to solve;
- describe the method you choose to solve this problem, and justify your choice;
- briefly describe your implementation, using pseudo-code;
- report the results of your experiments, and analyse them.

By the deadline of the assignment (for lab 1, it is September 16, 2005, at 5pm), you will send an email to the TA (his name and email will be posted on my website: http://www.cs.utep.edu/mceberio) with the following attachments:

- an archive of your java program (in format YourName.tar.gz);
- your report (in pdf format, YourName.pdf).

Problem 1. Parsing a text file

Consider again the exercise proposed in class. Implement the following algorithm:

- input: a text file, containing n words of maximum length l;
- output: the most frequent word along with its number of occurrences.

Perform experiments of your algorithm on different sizes of text files: in particular, the objective is to have n and/or l vary and to observe the changes in the number of steps performed. Use the experimental results you collect to confirm the results you got when working on the same problem as a homework.

Problem 2. Sudoku: a first approach

Please refer to http://en.wikipedia.org/wiki/Sudoku for a complete presentation of Sudoku.

1. Implement the following algorithm:
   - input: a n^2 by n^2 board totally filled with numbers from 1 to n^2;
   - output: true if the input board is a sudoku (i.e., satisfies all rules of the sudoku); false otherwise. *(cf. some examples below)*
2. Run your algorithm for different sizes of Sudokus (9 by 9, ..., n^2 by n^2).
3. Determine the complexity of your algorithm:
   - by proving it "on the paper";
   - by running experiments, reporting the results and analyzing them.

For instance, the following 4 by 4 board of sudoku is a solution. It satisfies all properties as described on the above-mentioned website.

1	2	3	4
3	4	1	2
2	1	4	3
4	3	2	1

On the other hand, the following one is not a sudoku.

1	2	4	3
4	3	1	2
2	1	4	3
4	3	2	1