CS161: Design and Analysis of Algorithms **Spring 2001**

Programming Project

Due: Tuesday, June 5th, 2001, 11:59pm.

1 Overview

For the programming project in this course you will design a set of data structures to emulate a database that stores a list of server names and corresponding IP addresses. Your goal is to achieve the best possible running time on worst-case input.

The operations your database needs to support are:

- **add** - add the server name/IP address pair to the database
- **delete** - given a server name, delete the corresponding pair from the database
- **lookup_ip** - given a server name, look up the corresponding IP address
- **lookup_server** - given an IP, look up the corresponding server name
- **find_next_used_ip** - find the smallest IP in the database greater than a given IP
- **find_prev_used_ip** - find the largest IP in the database smaller than a given IP

Each execution of your program will consist of a sequence of the above commands, distributed in one of two possible ways:

Command	Distribution 1	Distribution 2
add	25%	20%
delete	25%	20%
lookup_ip	25%	15%
lookup_server	25%	15%
find_next_used_ip	0%	15%
find_next_used_ip	0%	15%

For example, an input file of distribution type 1 will contain 25% add commands, 25% delete commands, and 25% of each type of lookup.

Prior to any commands being issued, your database will be given the distribution number for the input command sequence to be executed. To achieve best overall performance you should implement an optimized set of data structures for each of the two possible distributions, and depending on the distribution number choose the right one to use for processing the input commands.

2 Programming Environment and Honor Code

You may work in groups of up to three people. Only one submission per group is necessary.

Programming will be done on leland workstations. All code should be written in C. The makefile and skeletal files for reading input files and writing output files have been provided. You will only need to implement the appropriate functions in *db.c*, and modify the makefile to include extra modules, if any. The starer code is located in /usr/class/cs161/project.

You may not use any code written by anyone outside your project group other than the ANSI C libraries. Code written by your group members for courses taken in the past is allowed.

3 Grading

You will be graded on the worst-case running time of your program as well as your writeup. For each distribution we will generate a set of command sequences, time your program's execution for each one, and choose the largest (worst) running time among all the sequences for each of the two distributions. **The inputs will be specifically chosen to result in worst-case behavior in common data structures** (binary trees, hash tables, etc...), so be sure your implementation is resistant to adversarial input.

The overall project grade will be computed according to the formula $.4W + .3D_1 + .3D_2$, where W is the writeup grade and D_1 and D_2 are the grades given for the running-time performance on distributions 1 and 2, respectively. Programs which fail to compile or which crash during execution will receive no credit other than for the writeup. Programs which produce incorrect output will receive some credit for their running-time performance, but substantially less credit than programs which produce correct output, regardless of the relative speeds of the two programs.

4 Submitting

In addition to your source code, you should submit a README file containing:

- full name, leland username, and email address of each group member
- a high-level overview of the data structures used for each distribution
- an analysis of the asymptotic running time of each operation (for each distribution)
- a brief justification of major design choices
- analysis of the overall performance of your program, including miscellaneous "systems-level hacks" used to improve the running time
- any other comments about the assignment in general and your submission in particular

Remember, the writeup constitutes a considerable portion of your grade on this project and should be treated accordingly.

We will provide a script to submit your code and README electronically. This assignment is due at 11:59pm on Tuesday, June 5th. **No late submissions will be accepted.**