

## Data Structure and Algorithm II

### Homework #6 draft

Due: 10am, Monday, June 20, 2011

#### ==== Homework submission instructions ====

- Submit the answers for writing problems through the CEIBA system (electronic copy) or to the TA in R204 (hard copy). Please write down your name and school ID in the header of your documents. You also need to submit your programming assignment (problem 1) to the Judgegirl System(<http://katrina.csie.ntu.edu.tw/judgegirl/>).
- Each student may only choose to submit the homework in one way; either all as hard copies or all through CEIBA except the programming assignment. If you submit your homework partially in one way and partially in the other way, you might only get the score of the part submitted as hard copies or the part submitted through CEIBA (the part that the TA chooses).
- If you choose to submit the answers of the writing problems through CEIBA, please combine the answers of all writing problems into only one file in the doc/docx or pdf format, with the file name in the format of “hw5-[student ID].{pdf,docx,doc}” (e.g. “hw5\_b98902010.pdf”); otherwise, you might only get the score of one of the files (the one that the TA chooses).
- For each problem, please list your references (they can be the names of the classmates you discussed the problem with, the URL of the information you found on the Internet, or the names of the books you read). The TA can deduct up to 100% of the score assigned to the problems where you don't list your references.

#### ***Problem 1. The n-puzzle problem - the multithreaded version.***

In this problem, we ask you to re-write your program for HW5.1 - changing it to a multithreaded version.

OpenMP (Open Multi-Processing) is an application programming interface (API) that supports multi-platform shared memory multiprocessing programming in C, C++, and

Fortran on many architectures, including Unix and Microsoft Windows platforms. It consists of a set of compiler directives, library routines, and environment variables that influence run-time behavior. You may find lots of information about OpenMP here (including some simple examples): <http://en.wikipedia.org/wiki/OpenMP>. Please spend some time to familiarize yourself with it.

Please write a multithreaded program utilizing the OpenMP API to solve the n-puzzle problem. Note that your program should run faster than the original single-threaded program. Please also submit a report which includes a technical specification of your program, which helps the TA to understand your program. In addition, please answer the following questions in your report.

- Running your multithreaded algorithm on a computer with 4 cores, how much speedup would you expect to see? Please use the concepts you learned in the class (such as work, span, parallelism, etc.) to explain.

**Problem 2.** Give a multithreaded algorithm to multiply an  $n \times n$  matrix by an  $n$ -vector that achieves  $\Theta(n^2/\log n)$  parallelism while maintaining  $\Theta(n^2)$  work (problem 27.1-6).

**Problem 3.** Please solve problem 27-1 on page 805 of the textbook.