CISC106 Summer 2009 Lab05

- Review the code examples from class.
- Some programs below are associated with a question. **Answer the questions** using comments below your code in the m-file.
- The office hours of the TAs and the instructor are on the class website. Visit us!
- **NOTE:** Every function comment section should contain, at a minimum, **three examples** of the function being called and the result of evaluating the call. These examples must include boundary conditions (as discussed in class). Your test files must cover **at least** these exact examples (otherwise, why did you choose them?) and possibly more. Testing is important.

## **Problems**

1. Examine the selection sort function in Chapman 5.2., ssort.m (What do you think of the variable names? )

Use the test\_ssort.m function to verify that the selection sort is working. Now write a function that will generate a 1000x10 matrix with random values in each cell (you can use the rand function to do this).

Using the same test procedure that you used in Lab04, do some timing runs (at least 3) for the selection sort function using your randomly generated matrix (sort each row individually in a loop and time the overall loop).

2. Examine the merge sort function we did in class, mergesort.m.

Do the same timing runs on the same data as you did for selection sort. Do you notice any difference?

Now create a new random matrix for 1000x100 and 1000x1000 values and time both algorithms for all three matrixes. Plot each of these on a graph as in Lab04.

3. Sometimes combining algorithms can lead to performance benefits. Modify the mergesort.m to stop recursion earlier than length i= 1. Instead, use a function call to the ssort function to "solve the small problem". Find a value for the length to stop recursion that makes your new combined algorithm faster than both the original merge sort and selection sort functions. Plot your results.

If your TA requires a paper copy, be sure that you have a printed copy of your function M-files, script M-files, image files, and diary files demonstrating your testing. All must be stapled together, with your name and lab section on the top page.

Be sure that you upload a copy of all the MATLAB function, script, imasge, and diary files to Sakai. Then, click submit ONLY ONCE to send these to your Sakai and your TA.

On the first page of every printed copy for this course, your name, section, and TA's name must appear.