CSC503 Assignments for Winter 2012

Professor Massimo Di Pierro

February 1, 2012

General Ramarks

- These are individual assignments. You can discuss them with your colleagues but you cannot collaborate on the solution.
- Your solutions will be cross checked using turnit-in.
- Late assignments are not accepted under any condition.
- Proofs are valid only if they include steps, and explanations.
- Handwritten formulas are not allowed. You must Latex or Word equation editors. Latex preferred.
- All programs must be indented and commented
- The total is 100 points.
- You can get extra credit by helping reporting errors, bugs and typos in class notes.
- Points will be converted to letter grades based on this scale:

A	95-100
A-	92-94
B+	88-91
B	85-87
B-	82-84
C+	78-81
C	75-77
C-	72-74
D+	68-71
D	65-67
D-	62-64
F	0-55

0.1 Using and citing electronic sources

In conducting research for this course, I encourage you to consult those standard reference tools, scholarly projects and information databases, and peer-reviewed academic journals that may be found on the Internet in addition to traditional print resources. Keep in mind, however, that those electronic sources must be acknowledged. Please see the Modern Language Academy Handbook, section 4.9, for information on the correct citation of these sources.

0.2 Cheating and Plagiarism

Academic integrity entails absolute honesty in one's intellectual efforts. The DePaul Student Handbook details the facets and ramifications of academic integrity violations, but you should be especially aware of the policies on cheating and plagiarism. Cheating is any action that violates University norms or an instructor's guidelines for the preparation and submission of assignments. Such actions may include using or providing unauthorized assistance or materials on course assignments, or possessing unauthorized materials during an examination. Plagiarism involves the representation of another's work as your own, for example: (a) submitting as one's own any material that is copied from published or unpublished sources such as the Internet, print, computer files, audio disks, video programs or musical scores without proper acknowledgement that it is someone else's; (b) paraphrasing another's views, opinions or insights without proper acknowledgement or

copying of any source in whole or in part with only minor changes in wording or syntax even with acknowledgement; (c) submitting as one's own work a report, examination, paper, computer file, lab report or other assignment which has been prepared by someone else. If you are unsure about what constitutes unauthorized help on an exam or assignment, or what information requires citation and/or attribution, please ask your instructor. Violations may result in the failure of the assignment, failure of the course, and/or additional disciplinary actions.

0.3 About Computer programs

In this class, plagiarism includes submitting as your own work a computer program that was written by someone else, or directly derived from someone else. A program is directly derived from someone else's program if it is identical to someone else's program except for minor changes such as reformatting, change of variable names, etc.

1 Assignment 1: due Jan 23, 2012 (15 points)

Use the Psim simulator to implement a parallel version of ANY TWO sorting algorithms not covered in class, for example:

- Heapsort (8 points)
- Shellsort (7 points)

Exaplain each of the algorithms and discuss the running time on each of the network topologies defined in class. Provide benchmarks, including charts of T_p,S and E. (It is ok if p=2)

Reference: `http://www.inf.fh-flensburg.de/lang/algorithmen/sortieren/algoen.htm`

1.1 Extra Credit (5 points)

Reproduce this `http://www.tophatstuff.co.uk/?p=119` using python class `array`. Include code example and benchmark plots.

2 Assignment 2: due Fab 6, 2012 (15 points)

Use the Psim simulator to implement a parallel version of one of the following algorithms:

- BitonicSort
- RotateSort
- Even-odd mergesort

Explain the algorithm and discuss the running time of each of the network topologies defined in class. Provide benchmarks, including charts of T_p,S and E.

Reference: `http://www.inf.fh-flensburg.de/lang/algorithmen/sortieren/algoen.htm`

3 Assignment 3: due Feb 20, 2012 (15 points)

Implement the Rotate Sort ot the Bitonic Sort using MPI in C++.

4 Assignment 4: due March 10, 2012 (15 points)

Solve one of the following:

- If you have OpenCL use pyOpenCL to implement a sorting algorithm. Motivate your choice of the sorting algorithm.
- If you do not have OpenCL use MapReduce to implement a sorting algorithm. Motivate your choice of the sorting algorithm.

Discuss the running time and provide benchmarks, including charts of T_p,S and E.

5 Final Project: due Mar 17, 2012 (40 points)

Choose one of the following:

- Implement the PSim API on top of `mpi4py`
- Implement the PSim API so that processes can run on different machines and communicate via sockets (use the asyncore and asyncchat module like in mincemeat.py)
- Implement the the Parallel Prim algorithm using PSim or MPI
- Implement the the Parallel Kruskal algorithm using PSim or MPI
- Implement the the Parallel Dijkstra algorithm using PSim or MPI

You can propose a different project but you need to comminicate your choice before February 6.

You MUST use a version control system like GIT or Mercurial and make your code publicly available. Your code MUST be released under the BSD license. I will check your commit history.

For the final project you can work in groups of 2 or 3 students but you have to finalize your groups by February 6 and must include a report about the roles of each member of the group.