CSCI 3104: Algorithms
Homework 3

Due at **11:00am on Wednesday, September 23, 2015**. Submit your solutions electronically at moodle (name file as **LastName_FirstName_HW3.pdf**) or submit in paper before class. Also, submit your python source code electronically at moodle (name your file as **LastName_FirstName_HW3.py**). Make sure to include your name and student ID. Digital submission should also include the Honor Code Pledge (`http://honorcode.colorado.edu/about-honor-code`), and paper submission should include your signature indicating adherence to the Honor Code Pledge .

1. Suppose you are choosing between the following three algorithms:

   - Algorithm A solves problems by dividing them into four subproblems of half the size, recursively solving each subproblem, and then combining the solutions in linear time.
   - Algorithm B solves problems of size n by recursively solving two subproblems of size $n-1$ and then combining the solutions in constant time.
   - Algorithm C solves problems of size n by dividing them into nine subproblems of size $n/3$, recursively solving each subproblem, and then combining the solutions in $O(n^2)$ time.

   What are the running times of each of these algorithms (in big-O notation), and which would you choose?

2. Show that any array of integers $x[1, \ldots, n]$ can be sorted in $O(n + M)$ time, where

   $$M = \max_i x_i - \min_i x_i$$

   For small M, this is linear time!

3. A k-way merge operation. Suppose you have k sorted arrays, each with n elements, and you want to combine them into a single sorted array of kn elements.

   (a) Here is one strategy: Using the merge procedure from Chapter 2.3, merge the first two arrays, then merge in the third, then merge in the fourth, and so on. What is the time complexity of this algorithm, in terms of k and n?

   (b) Give a more efficient solution to this problem, using divide-and-conquer.

4. Write a python program that takes as input the name of a CSV file containing the historical quotes of a specific symbol (stock), utilizes the maximum subarray method (CLRS 4.1, Sep 11 lecture slides) to compute the optimal (i.e., maximum profit) buy and sale date pair based on the Open price for each day. Run your program on three

different stocks and report your results (stock symbol, start date, end date, buy date, sale date, maximum profit).

Go to `http://www.nasdaq.com/quotes/historical-quotes.aspx`, specify your symbol(s), press "Go". On the next page, you can change the Timeframe (default is 3 months), and at the bottom of the page, click "Download this file in Excel Format". Note that if you download the file when the stock market is open, the first row of quotes has a different format. You can simply remove that row from your downloaded file before testing.