Assignment 2

CS 350, Fall 2017
Due: October 17, 2017

Problem 1

(a) Design a recursive algorithm for computing $2^n$ for any non-negative integer $n$ that is based on the formula: $2^n = 2^{n-1} + 2^{n-1}$.

(b) Set up a recurrence relation for the number of additions made by the algorithm and solve it. Show your work.

(c) Is it a good algorithm for solving this problem? Describe a better approach.

Problem 2 In the programming language of your choice, implement both Insertion Sort and Merge Sort as described in class. Feel free base your code on implementations that you find online, just ensure that both operate on similar objects and properly cite your sources. Add code to both algorithms that tracks the time it takes to process an input. You will also need to create a method that generates arrays of integers of various sizes and orderings.

(a) Run your algorithms on various sized lists of random integers and graph the running time. Does your graph match the theoretical analysis we did in class, explain why or why not? (Try to get to the point where the slower of the two algorithms finishes in a little more than one second.)

(b) Repeat part (a) using already sorted lists of integers. Graph and explain your results.

(c) Choose a list size and experimentally determine what percentage of array elements can be out of order before Insertion sort is not longer the faster option. Graph and explain your results.