CS253 Programming Assignment 2

Fall 2003 (Due Thursday October 16, 2003) at the beginning of class

Be sure to read the Programming Style document when turning in your assignment. You must work in a small group for this assignment different from the last group you worked with.   The groups are also posted below.

This assignment continues assignment 1. Again we will use the actual machine to validate our run-time complexity analysis and asymptotic run-time complexity analysis by comparing worst case, best case, and average case run times for sorts, but now we will compare Merge Sort, Quicksort, and Heapsort.

To complete this assignment

- Determine, *a prior*i, which algorithm will have the biggest “leading constant” in its run time.
- You should implement all three algorithms, then plot their time behavior as a function of n and compare this with plots of the expected behavior.
- Determine n_0 for each algorithm as well as the leading constant.
- What inputs are required to generate “average” complexity. How about “best case and worst case”? Are the inputs the same for all algorithms? If not, is this a fair comparison?

To be effective in this assignment you should read and understand Chapter 6 thoroughly. Note that we will not discuss quicksort, specifically, in this class. Pseudocode is widely available. Additionally, asserts of invariants and required for merge sort and heap sort, but not for quicksort. It is possible to download all this code from site on the web, however, it’s much more difficult to debug and/or insert assert() statements into somebody else’s code. If you download the code from the web, it’s vital to provide the source.

Names	Group
James Townsend	4
Eric Mertens	1
Jeffrey Slane Jr	1
Benjamin Milster	1
Jonathan Mitchem	1
Kyle Owen	2
Esteban Aparicio	2
Benjamin Moss	2
Michael Ames	2
Shawn Tracy	3
Robert Pangrazio	3
Nguyen, Hai	3

Qian Fu Jim	3
Hoi Ho	4
Timothy Krupinski	4
Brian Van Vertloo	4