CSE331 Homework 2

(Assigned on January 30, 2017)

This homework contains two parts. The first part includes written problems. The second part involves programming.

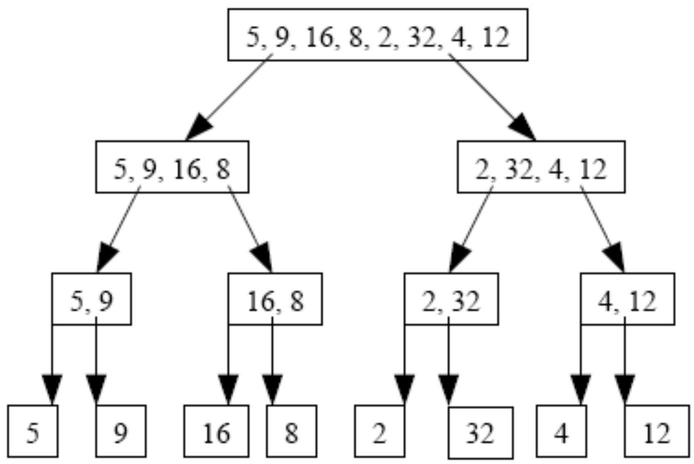
Written solutions to the written problems should be submitted as a printed or neatly handwritten writeup at the start of lecture on the due date. Alternatively, typeset electronic copies will be accepted via email to the instructor (kjl@msu.edu) prior to the due date; remember that non-typeset copies (e.g., scanned handwritten copies) are not eligible for electronic submission. See the syllabus for more information about late homework policies.

Only the source code(s) and README text file should be submitted via handin. Please make sure your program can be complied and run on arctic.cse.msu.edu, where we will test your programs. Due to the large class size, we cannot give credit to programs that cannot be run on arctic.cse.msu.edu.

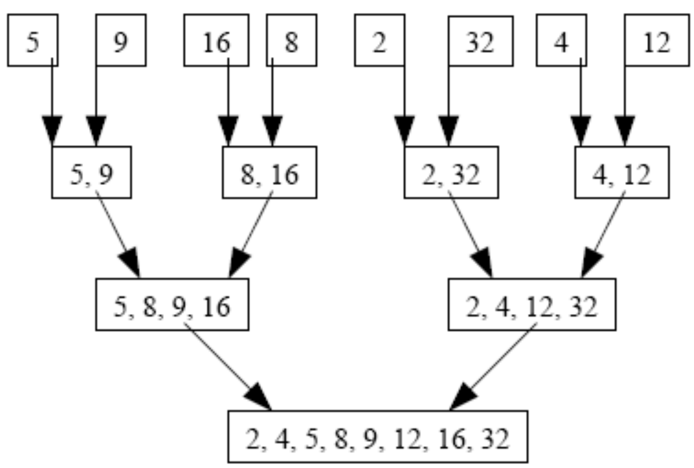
All problems are due at the start of lecture on **Febuary 6, 2017**.

Problem 1. Read the following example which shows how mergesort works. Then, follow the same procedure to sort {10, 99, 23, 12, 0, 5, 9, 8}. <10 points>

We begin with the unsorted list {5, 9, 16, 8, 2, 32, 4, 12}. Mergesort recursively breaks this list in half until it has formed lists with only one element in them. The following diagram shows the breakup occurring in parallel. In an actual program, of course, the splitting and merging steps would occur in sequence, but this is much easier to visualize.



Once the sequence has been broken down, the sublists are merged into sorted lists. The merging continues until a single sorted list has been generated from the input sequence.



Problem 2. What is the running time of insertion sort if all elements of the input array are equal? For this problem, you need to provide both the final answer ***and also the reasoning*** behind it. <5 points>

Problem 3. In class we solved the recursive equation, which is the running time of merge sort. When $T(n)=1+2T(n/2)+n$, we had $T(n)=n\log n+2n-1$. Following the in-class analysis, solve the recursive equation $F(n)=2F(n/2)+n$. The trivial case is still $F(1)=1$. <10 points>.

Problem 4. For this problem, implement quicksort and insertion-sort in any programming language (e.g., C, C++, etc.) and compare their running time for inputs with different sizes. A C++ framework can be found in the D2L course website under the Homework 2 tab in the Content section. The file is named compare-insertion-quicksort.cpp. You can use the random number generator and timing functions from this framework.
After testing the correctness of your sorting algorithms, compare and **report their running time** on inputs with following sizes:
10, 20, 100, 200, 400, 800, 1600, 3200, 6400

Submit your source codes via handin. The running time report should be submitted as part of the written solutions for this homework.
Correct implementation of insertion sort: 10 points, correct implementation of quick sort: 15 points, running time report: 10 points. If your program does not work correctly, we cannot give you credit.

Bonus problem: Watch the **Quick-sort with Hungarian (Küküllőmenti legényes) folk dance**
(https://www.youtube.com/watch?v=ywWBy6J5gz8). Write the pseudocode for the sorting algorithm presented in this dance. It is slightly different from the pseudocode introduced in class. Use the similar notation as the in-class pseudocode. <10 points>