CSCI 230 -- Lab 12

Minimum Spanning Trees (MST) & Heap Manager

Due:_____

Feel free to discuss and help each other out but does not imply that you can give away your code or your answers! Make sure to read all instructions before attempting this lab. You can work with a lab partner and submit one lab package for your group.

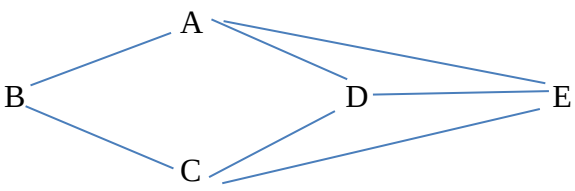
You must use an appropriate provided template from Canvas or my website (zeus.mtsac.edu/~tvo) and output "Author: Your Name(s)" for all your programs. If you are modifying an existing program, use "Modified by: Your Name(s)".

Lab question 1: What is a spanning tree?

Lab question 2: List some applications of MST.

Exercise 1: Implement one MST algorithm -- either Prim-Jarnik Algorithm or Kruskal Algorithm. Try a small graph below and print out the MST and total cost. Use the following weights:

- (A, B), 3
- (A, D), 5
- (A, E), 5
- (B, C), 4
- (C, D), 2
- (D, E), 5
- (C, E), 3



Exercise 2: Perform Project P-14.1 on page 687 of C++ book in C++ or Java. We will limit to only two of the four algorithms. Furthermore, you can limit your experiment as follow:

- Assume that you have an array of 1024 integers (4048 bytes or 4 KB) to manage
- Each allocation can be between 5 and 20 integers
- Start out with some allocations until about half of the memory is being used
- Perform some allocations and returns until an allocation cannot be made
- Tally up all available blocks (small blocks) to see how many blocks and how much memory left

Extra Credit: Implement the other MST algorithm (either Prim-Jarnik Algorithm or Kruskal Algorithm that is not done for regular assignment) or one more algorithm for exercise 2.

Online Submission: Submit one PDF file via Canvas includes status, answers to lab questions, output and source code for all required programs.