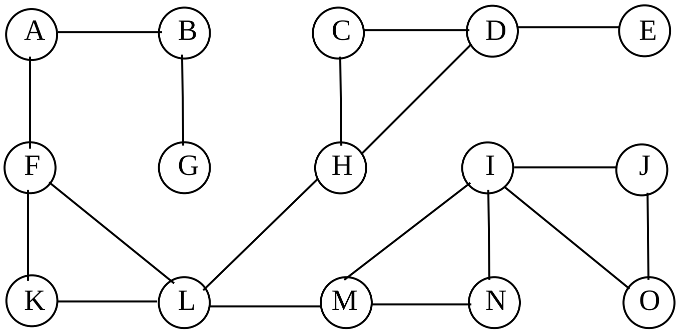
ECS 60 Written Homework #7: Articulation Points and Sorting (21 points) Fall 2015

Due Friday, December 4th, 4:00 pm in 2131 Kemper

1. (3 points) Given the following graph, using DFS, starting at A, searching in alphabetical order whenever multiple vertices may be processed, give each vertex a number followed by its low. Then list the articulation points. Your table should like the one below.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Vertex number (1 point)															
Low (1 points)															
Articulation Point Y or N (1 point)															

2. (8 points, 2 points each) Given the values 3, 1, 4, 8, 5, 9, 2, 6, 7 stored in an array in that order, provide the "states" of the array while being sorted for the sorting algorithms specified. Please write the contents of the array on a separate line for each state.
   a. For insertion sort, show the 8 states of the array as it is sorted by insertion sort, one state on each line. The "states" are achieved at the end of the for-loop when an entry has been moved to is currently proper position.

   b. Apply Shellsort using Shell's increment showing the 3 states of the array. The "states" are achieved at the end of the sorting based on each of the 3 increments.

   c. Apply heapsort to the array until the 8 and 9 are in the correct possitions. Show the 4 states of the heap as you use build heap when each sub-heap has been made into a heap, and then the two states of the array necessary until the 8 is in the proper position.

   d. Show the first 3 states that occur when applying quicksort to the array using median of 3, and cut-off of 2. A state occurs when quicksort has been applied to a subarray. There should be only one new pivot on each line. Underline the pivot used for each line. Apply the cut-off sort on the next line. When there are multiple subarrays to be sorted, choose to the subarray with the smaller values first.

3. (3 points) Weiss 7.43 "Suppose arrays *A* and *B* are both sorted and both contain *N* elements. Give an O(log*N*) algorithm to find the median of $A \cup B$.

4. (3 points) Weiss 7.45 "Suppose you have an array of *N* elements, containing three distinct keys`, true, false`, and `maybe`. Give an O(*N*) algorithm to rearrange the list so that all `false` elements precede `maybe` elements, which in turn precede `true` elements. You may use only constant extra space.

5. (4 points, 2 points each) Weiss 7.60 "There is a prize hidden in a box; the value of the prize is a positive integer between 1 and *N*, and you are given *N*. To win the prize, you have to guess its value. Your goal is to do it in as few guesses as possible; however, among those guesses, you may only make at most *g* guesses that are too high. The value *g* will be specified at the start of the game, and if you make more than *g* guesses that are too high, you lose. So, for example, if $g = 0$, then you can win in *N* guesses by simply guessing the sequence 1, 2, 3, …

   a. Suppose g = ceil(log *N*). What strategy minimizes the number of guesses?
   b. Suppose $g = 1$. Show that you can always win in $O(N^{1/2})$