June 5, 2007 Name ______________________________

EECS 311 Data Structures
Final Exam
Don’t Panic!

1. [10 pts] In the boxes below, show the result of adding the given numbers to a heap, using < as the predicate**.** Circle items that had to move at each entry.

1. After adding 10 to an empty heap.	2. After adding 12 to the previous heap.
3. After adding 1 to the previous heap.	4. After adding 14 to the previous heap.
5. After adding 6 to the previous heap.	6. After adding 5 to the previous heap.
7. After adding 8 to the previous heap.	8. After adding 15 to the previous heap.
9. After adding 3 to the previous heap.	10. After adding 9 to the previous heap.

June 5, 2007 Name ______________________________

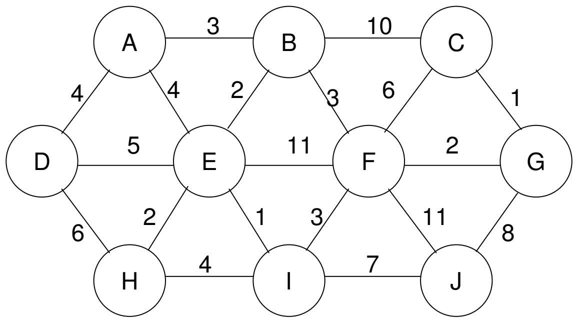
2. [10 pts] Show how Quicksort with median-of-three would sort the array below. Be very clear about what happens in each partitioning phase, e.g., for each phase, write something like:

> Partition __ to __, choose pivot __ from ________
> Swap __ with __, __ with __, __ with __, ...
> Resulting Array =

Be sure to include swaps done with the pivot at the beginning and end of a partitioning phase. Circle the items that are done and not involved in later phases. If a partition is 3 or fewer elements, just indicate the swaps needed, if any, to directly sort it.

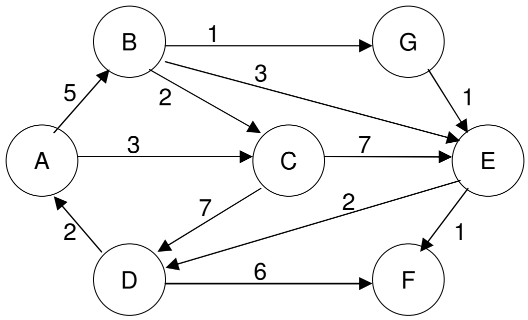
0	1	2	3	4	5	6	7	8	9	10
3	**1**	**4**	**0**	**8**	**9**	**2**	**6**	**7**	**-1**	**5**

3. [10 pts] Fill in the table below to show how Kruskal's algorithm would find a minimum spanning tree for the graph below. Draw a line on each edge of the graph used in the final MST. Is the MST unique? Justify your answer.



Edge	Weight	Action

4. [10 pts] Use the table below to show how Dijkstra's algorithm would find the shortest path from A to D. In the **Distance** column put the various distance values assigned to each vertex. In the **When Finalized** column put 1 for the first vertex that is finished, 2 for the 2nd vertex finished, etc.



When Finalized	Vertex	Distance
	A	
	B	
	C	
	D	
	E	
	F	
	G	

5. [15 pts total] a). Given an array of N elements of three different types: Cold, Warm, and Hot, design and describe clearly an O(N) in-place algorithm to put all the cold elements, on the left, followed by all the warm elements, followed by all the hot elements on the right. Your algorithm can use only a small constant amount of extra space.

b) Show how your algorithm would operate on this array:

0	1	2	3	4	5	6	7	8	9	10
C	**W**	**H**	**C**	**W**	**W**	**C**	**H**	**C**	**W**	**C**

c) Give as formal an argument as you can for why your algorithm is O(N) and correct for all possible arrays, even those with only a single kind of element.

6. [10 pts] Use dynamic programming to find the optimal solution to the sequence alignment problem below. A match scores 2 points, a mismatch scores -1, and a gap in either sequence scores -2.

Sequence 1: G A A T T C A G T T A
Sequence 2: G G A T C G A

		G	**A**	**A**	**T**	**T**	**C**	**A**	**G**	**T**	**T**	**A**
G												
G												
A												
T												
C												
G												
A												

Draw a line indicating a solution path in the table above, and show the actual alignment for that path below, using ─ to indicate any gaps. Is there more than one solution? Justify your answer.