

Name: \_\_\_\_\_

**CIS 23 – Exam 2**  
**Spring 2009**

\*\*\* You may choose *any* 5 out of the 6 questions on this exam. \*\*\*

For maximum partial credit, *show all your work* clearly. Be sure your final answers are clearly marked. You may work on the backs of the pages, but indicate that you are doing so.

**Question 1 (Selection)**

Assume you are given an array  $A$  with  $n$  distinct integers for which  $n > 3$ . Describe an efficient algorithm to solve the following two tasks.

- (a) Find an element in  $A$  that is neither the maximum nor the minimum.
- (b) Given an integer  $0 < k < n$ , find an element in  $A$  that is *not* one of the  $k$  largest elements in  $A$ . For full credit, your time complexity should be in terms of  $k$ .  
(Hint: you may want to use a similar idea to part (a) ).

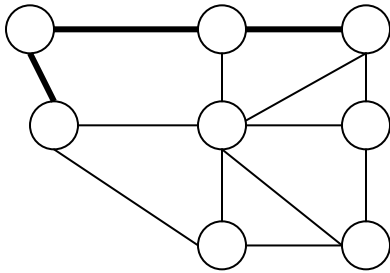
Use description in English, and perhaps illustrations, but do not write code.

**Question #2 (MST and Shortest Path)**

Part 1: We are running Prim's Minimum Spanning Tree algorithm on the following graph. We began with node A. The edges that are **bold** were already processed.

Which edge would be added next to Prim's MST? \_\_\_\_\_

After adding the edge, what is the list of fringe vertices: \_\_\_\_\_



Part 2: Show Dijkstra's shortest path tree for the graph in Part 1, beginning with node A.

***Question #3 DFS and BFS:***

- a) Perform a Depth-First Search of the following digraph. List the order in which vertices are visited, and show the data structure as it is used by the algorithm.
- b) Give the topological ordering of the nodes of the graph.
- c) Perform a Breadth-First Search of the digraph. List the order in which vertices are visited, and show the data structure, and its values, as the algorithm proceeds.

***Question #4 (QuickSort):***

(a) If you were running Quicksort on the following array, which element would be the *best* pivot? Why? Is that the pivot the algorithm chooses? Why or why not?

10 8 6 4 5 0 2 1 3 7 9

3

(b) Suppose that you chose the median of the first three elements of the array as the pivot. Would that change the worst case time complexity of the Quicksort algorithm? Why or why not?

(c) What was the advantage of using the partition algorithm that we saw in class (with the lo and high pointers)?

***Question #5 (Heaps and HeapSort)***

(a) Assume that you are given a heap with  $n$  nodes, in which every node is larger than its children except for exactly one node. In other words, the heap would be a max-heap, except that there is a single node that violates the maximality invariant. You do not know which node in the heap violates the max-heap property, and it may appear anywhere in the heap.

Show how you can fix this heap, by ONLY using calls to the procedure *fixheap*.

What is the worst case time complexity of your algorithm? Prove.

(b) Is heapsort an inplace sort? Explain.

## Question 6 (Graphs and their Representations):

**Part 1.** In an undirected graph,  $G=(V,E)$ , the *degree* of a vertex  $v$  refers to the number of vertices that are adjacent to  $v$ . Given the adjacency **matrix** representation of a graph, and a vertex  $v$ , describe an algorithm to compute the degree of vertex  $v$ .

Analyze the time complexity of your algorithm.

**Part 2.** In a directed graph,  $G=(V,E)$ , the *indegree* of a vertex  $v$  in  $V$  is defined as the number of edges coming into  $v$ , and the *outdegree* of  $v$  is the number of edges pointing away from  $v$ . Given the adjacency **lists** representation of a graph,  $G=(V,E)$ , and a vertex  $v$ , describe algorithms to compute the indegree and outdegree of vertex  $v$ .

Analyze the running time of your algorithms.