This is a set of sample exam questions for Exam 0. Some answers are given after ***.

1. Explain the difference between a Linked-List and an Array-List for the operations they both offer in terms of their Big-O runtime.

2. Using a min heap to implement a priority queue, show the state of the tree after the following operations:
   add(1)
   add(100)
   removeMin()
   add(4)
   add(3)
   add(1)
   add(2)
   removeMin()
   add(6)
   add(5)

   ***You need to draw trees but the heap will contain the following items at each step
   1
   1,100
   100 //1 REMOVED
   4,100
   3,4,100
   1,3,4,100
   1,2,3,4,100
   2,3,4,100 //1 REMOVED
   2,3,4,6,100
   2,3,4,5,6,100

3. Given two unsorted lists of characters with different lengths, write a pseudocode algorithm to combine and sort them in the most efficient way. Argue why your way is most efficient.

   ***Your answer will be order Big-O((n+m)log(n+m)) where n and m are the length of the lists.

4. What is the Big-Theta of the following program in terms of compares and also assignments.

```
min = (x1 - x2)^2 + (y1 - y2)^2
for i = 1 to n {
        for j = i+1 to n {
                d = (xi - xj)^2 + (yi - yj)^2
                if (d < min)
                        min = d
        }
}
```

   ***This program is Big-Theta(n^2) in terms of compares. i and j are assigned every iteration so it is similar to compares but we could also have a min update every time giving a total of 2n^2. Big-Theta(n^2) is also correct because a coefficient can bound it from above and below.

5. Given an array of integers, find three numbers which sum to 0. Write an O(n^2) algorithm in pseudocode for this problem. Example input / output:

   3-sum([4,-2,6,-3,-5,1]) == [4,-5,1]

6. Write an algorithm in pseudocode that will schedule jobs such that the maximum number of jobs are executed. Explain how the algorithm you use is greedy and how it is optimal. (start,end): (1,2),(3,5),(6,9),(2,3),(2,10),(1,3),(9,10)

7. In graph A, we would like to find the most central node. Write an algorithm in pseudocode that will compute the node with the shortest paths to all other nodes. You can use the function "sort(key[],val[])" which will return an array sort the values by the key array. You are given an array V[] which contains the nodes. For each node you can call functions like edges(V[2]) which will return another array.

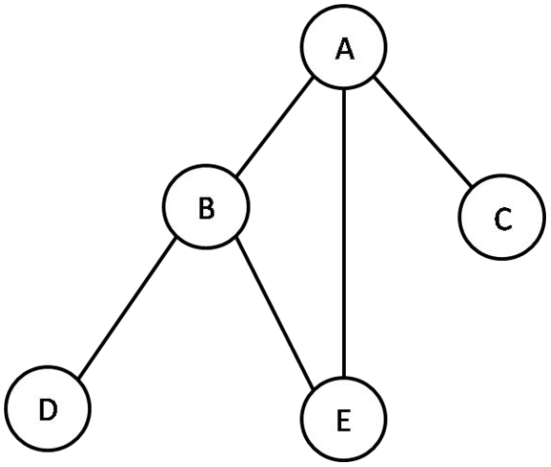
8. In graph B, compute the shortest path from node A to node D using Dijkstra's algorithm. Show the Priority Queue every time you visit a node.

   ***Shortest path is: A-E-D with a cost of 11

9. In problem 7, why won't breadth first search work?

   ***BFS doesn't take weight into account. A-B-D and A-E-D would have distance in terms of hops but different hops.

Graph A:



Graph B: