<u>Assignment 1</u>
<u>CSC 226</u>
<u>Theoretical Part</u>

<u>Question 1</u> (Searching, lower bounds). Argue convincingly: The problem of searching for the kth smallest element when given a sequence of n pairwise comparable elements has the following lower bound: Ω(n)
No matter under what circumstance, in order to determine any conclusions about a sequence of numbers, each element must be examined unless it is a sorted, or ordered sequence.

<u>Question 2</u> (Linear Selection). If we would vary the pivot determination in the linear selection algorithm studied in class such that we break the given sequence S in short sequences of size 3 instead of size 5 and then continue the algorithm as described, the running time is not linear anymore. Why? When writing up your answer, argue convincingly!
If the pivot was selected within sequences of size 3 instead of 5, there would have to be more pivot calls than necessary which slows down the algorithm substantially.

<u>Question 3</u> (MST, graph properties). In class, we studied the cycle and cut properties when determining minimum spanning trees for a given edge weighted graph. In both proofs we assumed that every edge in the given graph is of distinct weight.
a) Argue correctness of the cycle property for the case that edges in the graph can be of identical weight.
If C is any cycle in the graph G and we let 'e' be any edge in that cycle(because they are of identical weight), then the removal of any edge in that cycle will create a minimum spanning tree.

b) Argue correctness of the cut property for the case that edges in the graph can be of identical weight.
If there is an edge 'e' in graph G that has only one connection point, then by definition of a minimum spanning tree, it must be in the graph regardless of its weight.

<u>Question 4</u> (MST algorithm). Borůvka's algorithm assumes that all edges in the given graph are of distinct weight. Assume you are given a graph that does not satisfy this property but you are determined to apply Borůvka's algorithm to find a minimum spanning tree.
a) What problem can the algorithm run into? Give an example of a graph where Borůvka's algorithm as presented in class may fail.
You may run into a problem where you may have two different weights minimum spanning weights depending on which weighted edge is selected.

b) Expand Borůvka's algorithm to overcome the difficulty. Describe your algorithm in pseudocode.
When sorting the available weights within the graph, one way around having multiple edges of the same weight would be to visit each one incrimentally seeing which sequence produced the lowest weight spanning tree.