Assignment 2

CSC 226

Theoretical Part

Question 1. (Euclidean minimum spanning trees)

A **Euclidean minimum spanning tree** (EMST) is a **minimum spanning tree** of a set of n points in the plane, where the weight of each edge of the minimum spanning tree corresponds to each the distance between the two points corresponding to the edges end points.

In pseudocode, describe an algorithm that determines an EMST for a given point set in the plane.

Question 2. (Union-find data structure)

Consider the following sequence of union-find data structure operations on the set of elements {10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26} using union by rank and path compression.

Draw a figure to show the set(s) after all the operations are executed, indicating all parent pointer relationships and canonical elements. Also, indicate all the elements' ranks.

makeSet(10), makeSet(11), makeSet(12), makeSet(13), makeSet(14), makeSet(15), makeSet(16), makeSet(17), makeSet(18), makeSet(19), makeSet(20), makeSet(21), makeSet(22), makeSet(23), makeSet(24), makeSet(25), makeSet(26),
union(find(11),find(17)),
union(find(13),find(15)),
union(find(11),find(20)),
union(find(13),find(14)),
union(find(26),find(10)),
union(find(11),find(24)),
union(find(26),find(21)),
union(find(18),find(19)),

union(find(11),find(12)),
union(find(13),find(20)),
union(find(13),find(21)),
union(find(13),find(22)),
union(find(24),find(25))

Question 3. (AVL trees)

1. How many keys (internal nodes) does an AVL tree of height 9 have at least?
2. What is the maximum number of tri-node restructuring operations that needs to be executed to balance an AVL tree of height 9 after deleting a key? Give an example of such an AVL tree.

Question 4. (AVL trees) Determining x,y and z for restructuring after deleting a key from an AVL tree using binary search tree deletion is described as follows (as in lecture).

- Start at parent of removed node, walk up the tree. The first unbalanced node encountered is node z
- z's higher sibling is y
- y's higher sibling (if there is one) is x
- If both children of y are of the same height then:
  - if y is z's left child then x is y's left child
  - if y is z's right child then x is y's right child

In particular, the restructuring using these three nodes guarantees that the subtree restructured is an AVL tree after restructuring is done.

Why is this not guaranteed when using the following variant to determine x,y and z and therefore not a valid restructuring method?

- Start at parent of removed node, walk up the tree. The first unbalanced node encountered is node z
- z's higher sibling is y
- y's higher sibling (if there is one) is x
- If both children of y are of the same height then:
  - if y is z's left child then x is y's right child
  - if y is z's right child then x is y's left child

Justify your answer. Argue convincingly.