**Data Structure Midterm CPT**

1. Given an n by m matrix, transpose it to an m by n matrix using the fast transpose approach introduced in the lecture.
2. Implement sparse matrix multiplication.
3. Implement the failure function of the KMP algorithm to preprocess a given pattern *p*.
4. Invert a singly linked list.
5. Implement a stack with pop and push functions by using a linked list.
6. Implement an Infix to Postfix conversion function for an expression.
7. Evaluate a Postfix expression using a stack.
8. Implement a circular queue with insert and delete functions using an array.
9. Implement a binary search tree with insert and delete functions using a linked list.
10. Implement binary tree In-order, Post-order, and Pre-order traversals.
11. Implement a copy function to clone a binary search tree.
12. Given a sequence of n numbers, build a heap using an array and implement a heap sort that results in a sorted numbers in ascending order.
13. Given a sequence of n numbers, build a sorted list using a circularly linked list. Then, implement an insert function that inserts a given number to the list. The list should remain sorted after each insertion.
14. Given n equivalent relations, find the equivalent classes.
15. Given a 10 by 10 maze with one entrance and one exit, find the way to the exit (10,10) from the entrance (0,0). There are only four moving directions: right > down > left > up. You must use a stack to store the moving path.