**TC1018 – Data Structures**

**Fall 2018**

**Midterm exam – 75 minutes**

Instructions: Answer the questions below. You can use one sheet of notes

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

1. **[20 points]** Find the best answer to the following multiple choice questions
   1. What is the main characteristic of a stable sorting algorithm?
      1. The complexity is O(log n) consistently
      2. Elements with the same value appear in the output array in the same order as they do in the input
      3. Complexity is the same regardless of the input
      4. The sorting is done in place with O(n)
   2. What is a pointer in C/C++?
      1. Bit array
      2. Bytes
      3. Documents
      4. Memory Addresses
   3. Inserting an element to the beginning of an array (that is A[0] element) is more difficult than inserting an element to the beginning of a linked list.
      1. TRUE
      2. FALSE
   4. Suppose Node is a struct in C that contains two integer fields (4 bytes each) and a pointer (4 bytes) to the next Node. Consider the following instructions and select the:   
        **Node\* ptr = (Node\*)malloc(sizeof(struct Node));  
       Node\* tmp = ptr;**
      * 1. ptr = 4 bytes, tmp =8 bytes
        2. ptr = 8 bytes, tmp =2 bytes
        3. ptr = 12 bytes, tmp =4 bytes
        4. ptr = 4 bytes, tmp =12 bytes
        5. None of the above
   5. What is the best data structure to solve the following problem? A list needs to be built dynamically. Data must be easy to find, preferably in O(1). The user does not care about any order statistics such as finding max or min or median.
      1. Use an Array
      2. Use a Singly LL
      3. Use a Stack
      4. Use a Queue
      5. None of the above
   6. Consider a sorted circular doubly-linked list where the head element points to the smallest element in the list. What is the asymptotic complexity of finding the smallest element in the list?
      1. O(1)
      2. O(n)
      3. O(log n)
      4. None of the above
   7. Consider the same list from question *f*, what is the asymptotic complexity of determining whether a given element *e* appears in the list?
      1. O(1)
      2. O(n)
      3. O(log n)
      4. None of the above
   8. You want to build an address book with entries in alphabetical order by last name, what is the best data structure to do that?
      1. Singly Linked List
      2. Doubly-linked list
      3. Stack
      4. Binary Tree
      5. Graph
   9. You want to build a directory listing for an operating system, what is the best data structure option for this?
      1. Singly Linked List
      2. Doubly-linked list
      3. Stack
      4. Binary Tree
      5. Graph
   10. A stack is a FIFO structure and a queue is a LIFO structure
       1. True
       2. False
2. **[20 points]** Assume that LL is a DOUBLY linked list with the head node and at least one other internal node M which is not the last node. Write few lines of code to accomplish the following. You may assume that each node has a next pointer and prev pointer. You may NOT swap data to accomplish any of the following operations. For each operation, assume the original list as described above. You are encouraged to draw pictures to justify your code. Note that for each operation, you need to manipulate at least two pointers, next and prev.
   1. [5 pts]Delete the head node
   2. [10 pts]Insert a node P immediately after M

* 1. [15 pts] Swap head and the node M (you may not swap data)

1. [10 pts] Implement the find(int n) method in a given circular singly-linked list.

1. [15 pts] Implement the printInReverse() function for a given Linked List, **recursively**.
2. [15 pts] Implement a function to insert a new node with given key in a binary search tree
3. [20 pts] Write a function that converts a binary search tree to a sorted doubly linked list.  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   EXTRA CREDIT: Implement a function to check if a tree is balanced. For the purposes of this question, a balanced tree is defined to be a tree such that no two leaf nodes differ in distance from the root by more than one.

RESCUE [5 points] Name of the recently deceased Microsoft co-founder.