LAST NAME (PRINT): ____________________ FIRST NAME (PRINT): ____________________

CS 367: Introduction to Data Structures

Midterm Exam 2
Exam code: A0

Friday, August 11th 2017

90 points (26% of final grade)

Instructor: Meena Syamkumar

1. **Fill in these fields and their bubbles on the scantron form (use #2 pencil).**
   (a) LAST NAME - fill in your last (family) name starting at left column.
   (b) FIRST NAME - fill in first five letters of your first (given) name.
   (c) IDENTIFICATION NUMBER is your UW Student ID number.
   (d) SPECIAL CODE is your exam code
2. **DOUBLE-CHECK THAT YOU HAVE FILLED IN THE BUBBLES FOR EACH COLUMN OF ABOVE FIELDS ON SCANTRON.**
3. **Read, agree to, and sign this ACADEMIC CONDUCT STATEMENT.**

   I will keep my answers covered so that they may not be viewed by another student during the exam or prior to completion of their exam. I will not view or in any way use another's work or any unauthorized devices. I understand that I may not make any type of copy of any portion of this exam. I understand that being caught doing any of these or other actions that permit me or another student to submit work that is not our own will result in automatic failure of the course. All such penalties are reported to the Deans Office for all involved.

Signature:

Parts	Number of Questions	Question Format	Possible Points
I	15	Simple Choice	15
II	15	Multiple Choice	45
III	6	Written	30
	36	Total	90

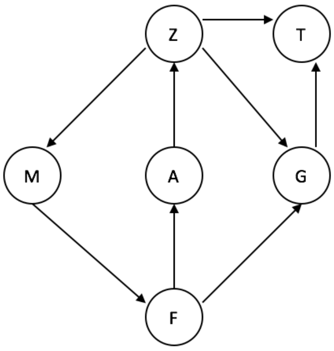
Turn off and put away all electronic devices and wait for the proctor to signal the start of the exam.

Part I: Simple Choice (1 point each)

Select the word or phrase that makes the statement **true**.

If there is no ______________, then mark **A** if the statement is true or **B** if the statement is false. Be sure to mark the corresponding letter on the answer sheet (scantron). Unless otherwise specified, assume the ADTs, data structures, interfaces, and algorithms mentioned are those discussed in lecture and in the readings.

For the following 3 questions consider the below graph:



1. The graph is *directed.* A. *True* B. *False*
2. The graph is *acyclic.* A. *True* B. *False*
3. The graph is *weakly connected.* A. *True* B. *False*
4. A *pre-order traversal* on a general tree visits the nodes in the same order as:

   A. a *depth-first search* starting at the root of that tree

   B. a *breadth-first search* starting at the root of that tree

5. Which of the following has a lower complexity to sort an array of N objects that is *already in sorted order?*

   A. *merge sort*

   B. *quick sort* using the first item in each part as the pivot

Part II: Multiple Choice (3 points each)
Choose the best answer of the available choices.

6. Consider implementing a hash table to be used to store **20-digit integer keys**. Which one of the following **hash functions** is *most likely* to return **different hash indexes** for two different keys?

   A. *A hash function based on the first 5 digits of the key.*

   B. *A hash function based on the last 5 digits of the key.*

   C. *A hash function based on the product of the digits of the key.*

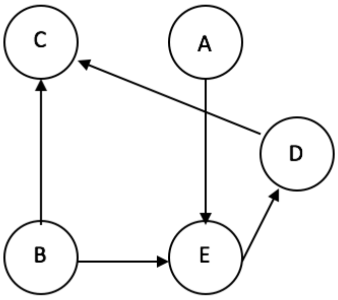
   D. *A hash function based on the sum of the digits of the key.*

   E. *A hash function based on the sum of two numbers: one number contains the first 10 digits of the key and the second number contains the last 10 digits of the key.*

7. Assume *linear probing* collision resolution is used on a **hash table** of size 11 having a hash function that is simply the key *mod* by the tables size. Which one of the following shows the hash table that results from inserting these keys in the order given: 26, 13, 6, 15, 19, 4, 16, 22
Note: "|" denotes an array element separator here

   A. | 22 | | 13 | | 4 | 16 | 6 | 26 | 19 | 15 | |
   B. | 22 | | 13 | | 26 | 15 | 6 | 4 | 19 | 16 | |
   C. | 22 | | 13 | | 4 | 16 | 6 | | 19 | | |
   D. | 26 | 13 | 6 | 15 | 19 | 4 | 16 | 22 | | | |
   E. | 22 | | 13 | 4 | 26 | 15 | 6 | | 19 | 16 | |

8. Consider the following ordered lists of nodes and the **directed graph**:



Which of following lists the nodes in *topological order* for the graph above?

   i. A B C D E
   ii. A B D C E
   iii. A B E D C

   A. *i only*
   B. *ii only*
   C. *iii only*
   D. *ii and iii only*
   E. *i, ii, iii*

9. Given the following array:

| 8 | 3 | 5 | 9 | 7 | 1 | 0 | 6 | 4 | 2 |

Which one of the following represents what the array looks like after **three passes** (i.e., three iterations of the outer loop) of the **bubble sort** algorithm as covered in lecture?

   A. | 0 | 1 | 2 | 3 | 5 | 4 | 6 | 7 | 8 | 9 |
   B. | 0 | 1 | 2 | 8 | 3 | 5 | 7 | 9 | 4 | 6 |
   C. | 0 | 1 | 2 | 8 | 3 | 5 | 9 | 7 | 4 | 6 |
   D. | 0 | 1 | 2 | 8 | 3 | 5 | 9 | 7 | 6 | 4 |
   E. | 0 | 1 | 2 | 9 | 7 | 3 | 8 | 6 | 4 | 5 |

10. Assume that you have the following *incomplete* defintions of *Graph* and *Graphnode* classes:

```
class Graphnode<T> {
        private T data;
        private int index ; // for indexing into edge matrix
           or nodes array
        public T getData() { return data ; }
        public int getIndex() { return index; }
}
```

Consider writing the following method which calculates the in-degree of a given node n. An incomplete version of the method is as follows:

```
private int inDegree(Graphnode<T> n) {
        int nodeNUm = n.getIndex(); // for indexing into edge
           matrix
        int deg = 0;
        for (int i=0; i < nodes.length; i++ )
                STMT
        return deg;
}
```

Which of the following replacements for *STMT* correctly completes this method?

A. `if (edge[i][nodeNum]) deg++;`

B. `if (edge[nodeNum][i]) deg++;`

C. `if (edge[i][nodeNum]) deg++;`
   `if (edge[nodeNum][i]) deg++;`

D. `if (edge[i][nodeNum] || edge[nodeNum][i]) deg++;`

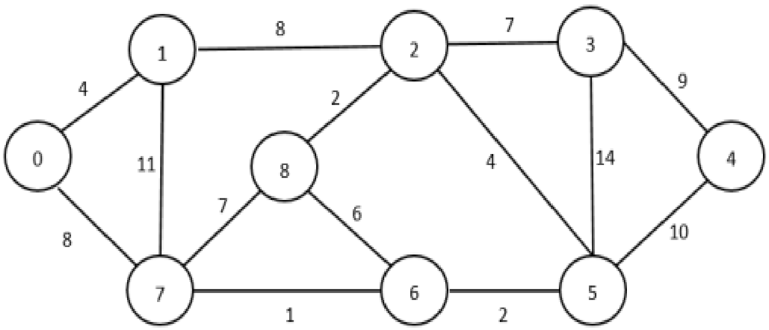
E. `if (edge[i][nodeNum] && edge[nodeNum][i]) deg++;`

Part III: Written (5 points each)
Write your answer within the provided space as briefly as possible.

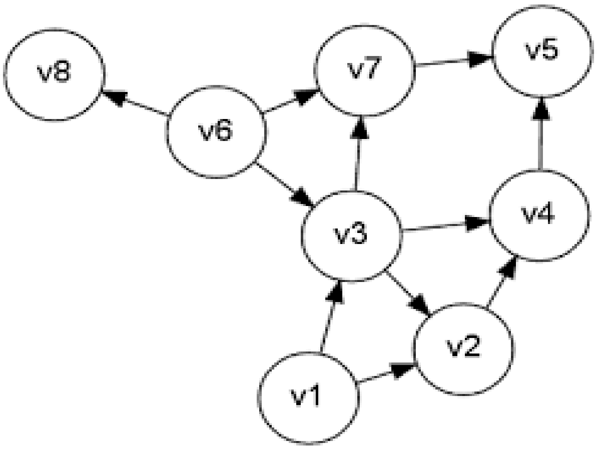
11. Show the **red-black tree** that results from inserting the following sequence of integers into a tree that is **initially empty**.
Show the *black nodes* using circle and *red nodes* using squares as in lecture. It is not mandatory to use the specific color pens for these.

```
41, 53, 65, 44, 43, 52, 49, 68, 45, 39
```

12. Consider the following **undirected graph**.
Trace **Dijkstra's algorithm** starting at node **1**.



13. Show the **topological ordering** for the graph below using the iterative method covered in lecture (rather than the recursive method in the readings).
Use the 367 convention of choosing vertices in alpha-numeric order.
Show the **Stack contents** while tracing through the iterations and also show the assignment of **num** as shown in the lecture example. Finally, show the **valid topological order** obtained.



14. Consider the following **undirected graph**
    Use the 367 convention of choosing vertices in alpha-numeric order.

    A. Trace the **BFS traversal** for this graph starting at node **0**.
    B. Trace the **DFS traversal** for this graph starting at node **0**.