CMSC132 Summer 2017
Final

First Name (PRINT): ____

Last Name (PRINT): ____

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Your signature: ____

Instructions

- This exam is a closed-book and closed-notes exam.
- Total point value is 100 points.
- The exam is a 120-minutes exam.
- Please use a pencil to complete the exam.
- **WRITE NEATLY**. If we cannot understand your answer, we will not grade it (i.e., 0 credit).

Grader Use Only

#1	Fill in the Blanks	15	
#2	Multiple Choice	25	
#3	Short Answer	36	
	Coding Problems	24	
Total	Total	(100)	

1. [15 pts, 1 pts each] Fill in the blanks

Fill in the worst-case time complexity for the following algorithms

1) Selection sort O( )
2) Merge Sort O ( )
3) Heap sort O( )
4) Insertion Sort O( )
5) Finding a key from a binary tree O( )
6) Finding a key from a binary search tree O( )
7) Finding a key from a Red Black Tree O( )
8) Finding a key from a linked list O( )
9) Finding a key from a 2-3-4 tree O( )
10) Finding a key from a no-collision Hash table O( )
11) Hanoi tower Problem (n is the number of disks) O( )
12) Dijkstra's shortest path algorithm (without priority queue) O( )
13) Insert a key into a binary heap. O( )
14) Binary search of a sorted array O( )
15) Push, pop, peek operations on linked list based stack O( )

2. [25] Multiple Choice Questions

1) [2] What is the value of mystery(4)?

```
public static int mystery(int n){
   if(n == 2) return 2;
   else return 3 * mystery(n-1);
}
```

   a) 27
   b) 36
   c) 18
   d) 24

2) [2] What is the complexity of the following code in terms of **n**?

```
int sum = 0;
for (int i = 0; i < n; i++)
    sum += i;
for (int j = 0; j < n; j++)
    for (int k = 0; k < 100; k++)
         sum += j*k;
```

   a) O(n)
   b) $O(n^2)$
   c) $O(n^2 \log n)$
   d) $O(n^3)$

3) [2] Which in the following is a pre-order traversal of a **valid** BST?
   a) 11, 17, 2, 7, 20, 25, 4
   b) 11, 4, 2, 7, 17, 20, 25
   c) 1, 7, 4, 25, 20, 17, 11
   d) Both a, b are valid

For the following 3 questions, use this Node class definition

```
public class Node
{
   private int data;
   private Node next;
   ...
}
```

4) [2] Suppose cursor refers to a node in a linked list (using the Node class). What statement changes cursor so that it refers to the next node?

```
a) cursor++;
b) cursor = next;
c) cursor += next;
d) cursor = cursor.next;
```

5) [2] Suppose cursor refers to a node in a linked list (using the Node class). What boolean expression will be true when cursor refers to the tail node of the list?

```
a) (cursor == null)
b) (cursor.next == null)
c) (cursor.data == null)
d) (cursor.data == 0.0)
```

6) [2] Which boolean expression indicates whether the numbers in two nodes (p and q) are the same. Assume that neither p nor q is null.

```
a) p == q
b) p.data == q.data
c) p.link == q.link
d) None of the above.
```

7) [2] Which of the following applications may use a stack?

a) Parentheses balancing program.
b) Keeping track of local variables at run time.
c) Syntax analyzer for a compiler.
d) All of the above.

8) [2] Suppose T is a binary tree with 14 nodes. What is the minimum possible height (number of edges in the longest path) of T?

a) 0
b) 3
c) 4
d) 5

Use this binary tree for questions 9-11

```
        14
      /    \
     2      11
    / \    /  \
   1   3  10   30
         /    /
        7   40
```

9) [2] What is the order of nodes visited using a pre-order traversal?

a) 1 2 3 7 10 11 14 30 40
b) 1 2 3 14 7 10 11 40 30
c) 1 3 2 7 10 40 30 11 14
d) 14 2 1 3 11 10 7 30 40

10) [2] What is the order of nodes visited using an in-order traversal?

   a) 1 2 3 7 10 11 14 30 40
   b) 1 2 3 14 7 10 11 40 30
   c) 1 3 2 7 10 40 30 11 14
   d) 14 2 1 3 11 10 7 30 40

11) [2] There is a tree in the box at the top of this section. What is the order of nodes visited using a post-order traversal?

   a) 1 2 3 7 10 11 14 30 40
   b) 1 2 3 14 7 10 11 40 30
   c) 1 3 2 7 10 40 30 11 14
   d) 14 2 1 3 11 10 7 30 40

12) [2] Which of the following points is/are true about Linked List data structure when it is compared with arrays

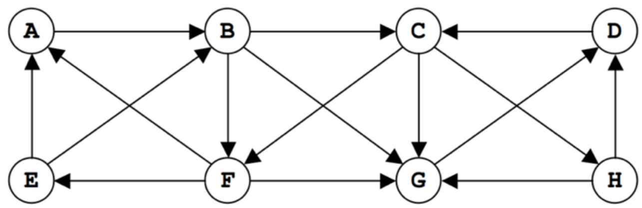
   a) Arrays have better cache locality.
   b) Easy to insert and delete elements in Linked List
   c) Random access is not allowed in Linked Lists
   d) All of the above

13) [1] To start the execution of a thread after you create it, you
   a. Call the run() method
   b. Call the start() method

.

3. [36] Short Answer

1) [4] Run recursive depth-first search on the digraph below, starting at vertex A and print the vertices in the order they are processed by DFS. As usual, assume the adjacency lists are in lexicographic order, e.g., when exploring vertex F, the algorithm considers the edge F→A before F→E or F→G.



Answer:

2) [2] Fill in the array after the FIRST iteration of the large loop in a selection sort (sorting from smallest to largest).

Original	5	3	8	9	1	7	0	2	6	4
After 1 iteration										

3) [2] Give two different reasons to explain why the following binary tree is not a max-heap:

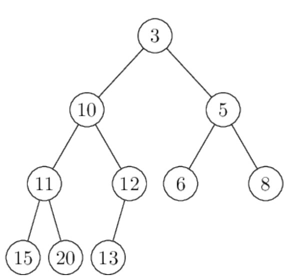
```
    91
   /  \
  77   46
 / \     \
68  81    11
```

Answer:

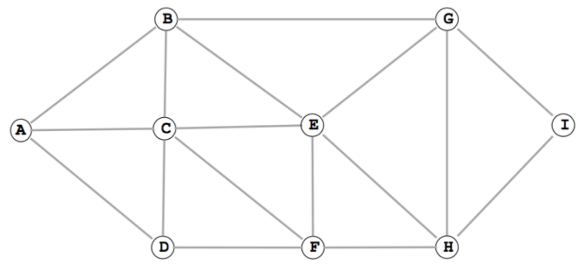
1)

2)

4) [3] Draw a new heap that is created by removing one item (removes root) from the following min-heap:



5) [4] Run breadth-first search on the graph below, starting at vertex A. As usual, assume the adjacency sets are in sorted order, e.g., when exploring vertex F, the algorithm considers the edge F-C before F-D, F-E, or F-H.



Fill in the vertices in the order in which the vertices are visited.

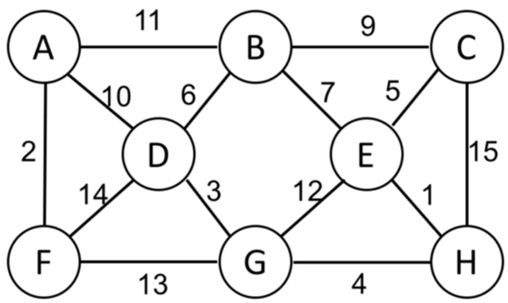
A	B							

6) [4] Construct the binary tree given its inorder and post-order traversal as follows:
In-order: 6,2,1,5,7,8,4,3
Post-order: 2,6,5,1,4,3,8,7

7) [4] Use the hash function H(K)=K mod 7, insert these integer keys into the hash table: 21,55,46,39,40,32

0	1	2	3	4	5	6

8) Consider the following weighted. Note that the edge weights are distinct integers.



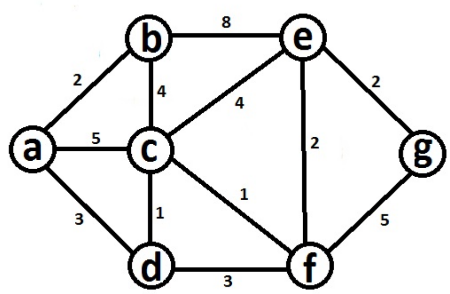
a. [3] Complete the sequence of edges (fill in the weight) in the MST in the order that Kruskal's algorithm includes them.

1	2	3						

b. [3] Complete the sequence of edges in the MST in the order that Prim's algorithm includes them. Start Prim's algorithm from vertex A.

2	10	3						

9) Run Dijkstra's algorithm on the weighted digraph below, starting at vertex A.



a. [4] List the vertices in the order in which the vertices are deleted from the priority queue and give the length of the shortest path from A.

Vertex	a	b						
Distance	0	2						

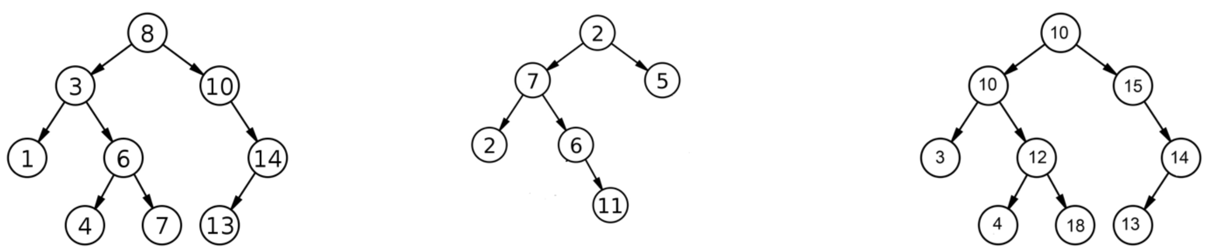
b. [3] Draw the edges in the shortest path tree with thick lines in the graph above.

4. [24] Programming

```
class Node{
      Integer data;
      Node left,right
      Node(Integer i){ data = i;}
}
```

1) [9] Given the Node class definition above, write a function "**`Node merge(Node t1, Node t2)`**", which receives two binary trees and returns a **new** tree by merging the nodes in the same location in two given trees by adding their keys. If one tree is empty, copy the other tree.
   For example: **`Node t3 = merge(t1,t2)`**

t1 t2 t3



2) [9] Write the iterative version of the binary tree preorder traversal. function “**`void iterativePreorder(Node t1)`**” receives a binary tree and prints the keys in preorder traversal order. You are not allowed to use recursive calls.

3) [6] Write a recursive function “**`int count(ListNode head, int key)`**”, which returns the number of occurrence of the key in the given linked list. Head references the first node in the list.

```
class ListNode{
      int data;
      ListNode next;
}
```