CS2230 Data Structures
Final Exam

December 16, 2016
7:30-9:30AM

Total points = 200

Name _____

Last three digits of id _____

Please turn off your cell phones.

This is a closed book closed note closed Internet test.

There are ***eight pages*** and ***seven questions***. You can use calculators.

You can bring one 8.5 x 11" sheet (front and back) with notes of your choice written on it

Q1. (5x10= 50 points)

Part (a) Both *hash tables* and *binary search trees* are used for storing key value pairs. Hash tables promise search, insert and delete operations in constant time, whereas binary search trees need O(log n) time for the same operations. Why would anyone use binary search trees instead of hash tables?

Answer.

Part (b) Suppose that the Java library `java.util.LinkedList` is implemented using a *doubly linked list*, maintaining a reference to the `first` and `last` node in the list, along with its `size`.

```
public class LinkedList<Item> {
      private Node first; // the first node in the linked list
      private Node last; // the last node in the linked list
      private int N; // number of items in the linked list

      private class Node {
            private Item item; // the item
            private Node next, prev; // next and previous nodes
      }
      ...
}
```

What are the best estimates of the worst-case running times of the following operations in big-O notation? (Choose among O(1), O(log N), O(N), O(N log N))

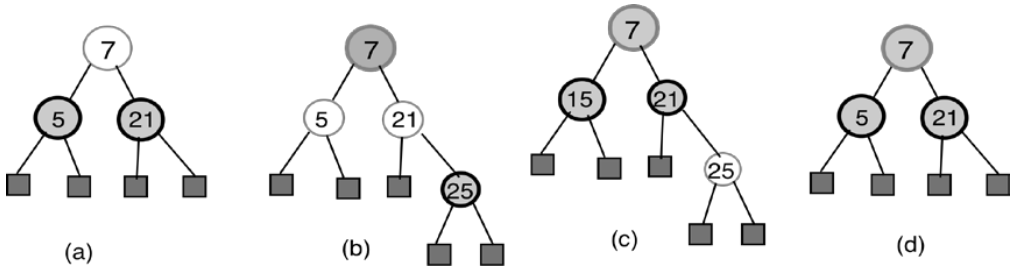
		Answer
`addFirst(item)`	Add the item to the beginning of the list	
`get(i)`	Return the item a position i of the list	
`Set (i, item)`	Replace position i of the list with item	
`removeLast()`	Delete and return the item at the end of the list	
`contains(item)`	Is the item in the list?	

Part (c) The height of a *Binary Search Tree* (BST) depends on the order in which the keys are inserted into an empty tree if no balancing operation is performed.

Given an initially empty BST, in what order will you insert the keys A, B, C, D, E, F, G so that the height of the BST is a minimum. Use the natural alphabetic order to sort the keys, i.e., A < B < C < D < E < F < G.

Answer.

Part (d) Which of the following four trees (a) through (d) are Red Black trees? Here, the black shaded circles denote black nodes and the clear circles denote red nodes. Note that the square shaded boxes are terminal nodes that are always black.



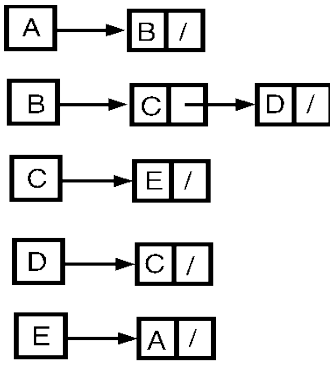
Answer.

Part (e) What is the best estimate of the time complexity of the following method using the big-O notation?

```
public static void f(int N) {
      int sum = 0;
      for(int i = 1; i <= N; i = i*2) {
            for (int j = 0; j <= N; j=j+1) {
                  sum = sum + 1
      }
   }
}
```

Answer.

Q2 (20+10 =30 points). The adjacency list representation of a graph with five vertices A, B, C, D, E is given below.



/ = null symbol

Part (a). Draw the adjacency matrix (complete the partial drawing below), and

Answer

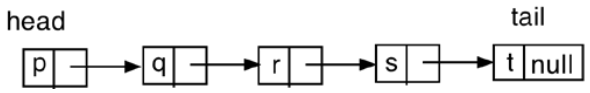
Part (b). Draw the graph.

Answer

Q3 (20 points). Consider a *singly linked list* of nodes and the following method is called on the head node of the list:

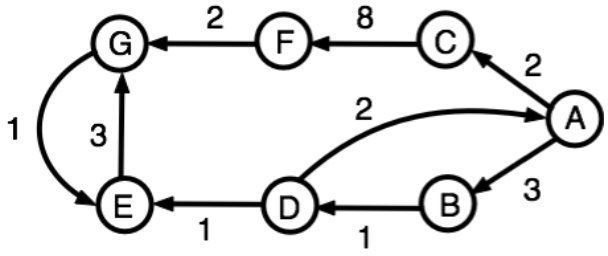
```
void mystery(Node<T> node) {
        if(node == null)
                return;
        mystery(node.getNext());
        System.out.println(node.getData());
}
```

The given linked list is shown below. What will the above program print?



Answer:

Q4 (20 points) Consider the following directed graph.

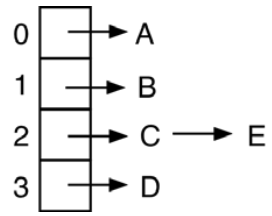


Write down the order in which the vertices will be visited by a *depth first search* starting from the node A. If there are multiple valid choices, then assume that the alphabetically earlier vertices will be visited first (i.e., if both C and E are valid choices in a certain step, the choose C since C < E).

Answer.

Q5. (30 points)

Five keys A, B, C, D, E with hash codes 8, 25, 10, 15, 18 respectively have been added to a hash table of size 4 that uses Separate Chaining (see below).



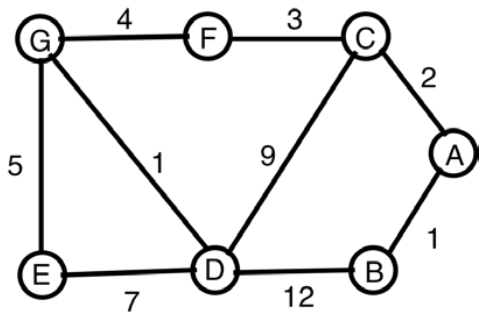
You have to insert one more key F with hash code 43 into the hash table.

Note that the implementer of the hash table decided to *resize the table* by doubling its size when the *load factor* reaches or exceeds 1.5. Draw the new hash table after we insert F.

Answer.

Q6 (20+10=30 points)

In the graph below, use Dijkstra's shortest path algorithm to compute the shortest paths from vertex A to all the other vertices. Your answer will consist of the following two parts:



Part (a) Starting from vertex A, write down the order in which the vertices will be added by *Dijkstra's shortest path algorithm* while computing the shortest paths.

Answer.

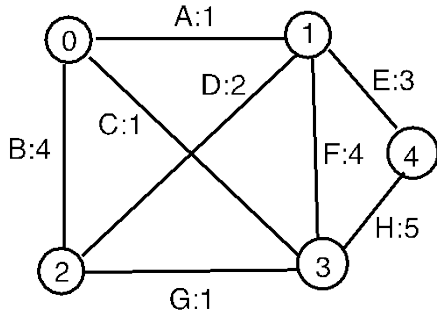
Part (b) Draw the *shortest path tree* with vertex A as the root. The *shortest path tree* will consist of the edges that belong to the shortest paths from A to every other node.

Answer.

Q7. (20 points)

For the graph below, use Kruskal's algorithm to find the Minimum Spanning tree. The number on each edge denotes its weight, and the letter is a unique label you should use in your answer to specify that edge. Provide the edges in the order in which they would be found by Kruskal's algorithm. Break any ties using the alphabetical label.

Give your answer in terms of the alphabetical labels, e.g. if Kruskal's algorithm adds the edge between vertices 3 and 4, you would write H for that edge



Answer.