CSC130 Final Practice Exam

Spring 2018

1. (10 Points) Find the best 1-1 mapping between column A and column B.

A	B
1) Greedy Algorithm	a) Merge Sort
2) Divide & Conquer	b) Prim's MST algorithm
3) Randomized Algorithm	c) Ordering Matrix multiplication algorithm
4) Dynamic Programming	d) Tic-Tac-Toe Minimax Algorithm with α-β pruning
5) Backtracking	e) Skip Lists

2. (10 Points) Answer the following questions and provide brief explanation.

1) What is the maximum number of edges in a graph with V vertices and no parallel or self-loop edges?
2) If a graph has V vertices and E edges, how much space is required for representing that graph using adjacency list in big-O notation?
3) Given V is the number of vertices and E is the number of edges in an undirected graph, what is the average degree?
4) Explain why resizing is necessary for symbol table implementations using hashing with linear probing.
5) Given n integers in the range 0 . . . k, describe how to preprocess these integers into a data structure that can answer the following query in O(1) time: given two integers a and b, how many integers fall within the range a . . . b?

3. (10 Points) Mark each of the following statement true or false.

True/False	Statements
	Suppose that a graph has distinct edge weights, its shortest edge has to belong to the MST (Minimum Spanning Tree).
	Adding a constant to every edge weight does not change the solution to the single-source shortest-paths problem for an edge-weighted digraph.
	The depth of any node in a forest built by weighted Quick-Union for N sites is at most lgN.
	Network Max-Flow problem is a NP-Complete problem.
	If a node in a BST has two children, its successor has no left child and its predecessor has no right child.

4. (10 Points) Provide short answers for the following questions in big-O notation. Assuming n is the number of elements in the symbol table.

1) For unordered linked list symbol table implementation, what is the worst case running time for search?

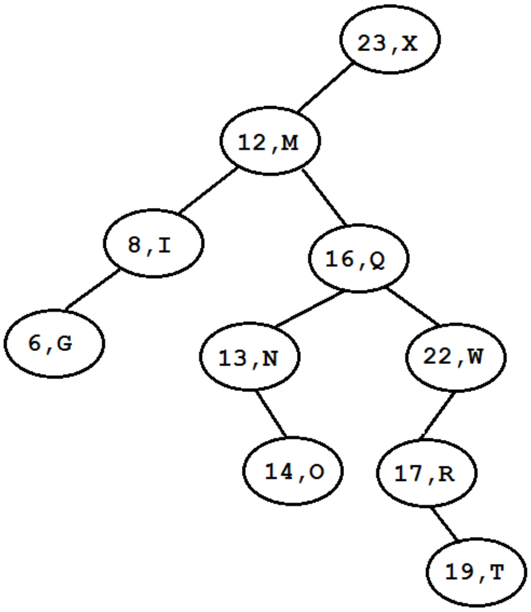
2) For ordered array symbol table implementation, what is the worst case running time for search?

3) For binary search tree based symbol table implementation, what is the worst case running time for search?

4) For red-black tree based symbol table implementation, what is the worst case running time for search?

5) For symbol table implementation based on hashing with separate chaining, what is the average-case running time for search? Assume the implementation has a good hash function and a resizing policy to keep lists short.

5. (10 Points) Given the following BST where key value pair is shown on each node. Fill in blanks in Column B.

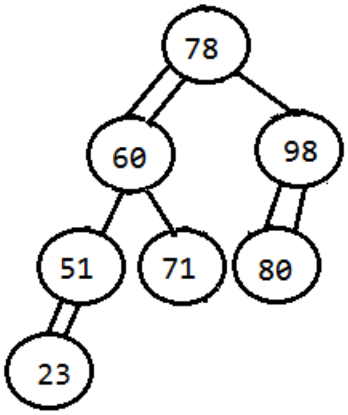


A	B
min() // smallest key	
max() // largest key	
rank(17) // number of keys less than the key	
select(6) // key of rank k	
floor(20) // largest key less or equal to the key	

6. (10 Points)

1) Given the following keys `[83, 43, 18, 39, 59, 25, 78, 35, 36, 92], insert` the keys in the provided order into an initially empty BST and provide the final BST after all the insertions. Only key needs to be displayed for each BST node.

2) Given the following red-black BST, where red link is represented by double lines and black link is represented by single line. Provide the final result after inserting 11 into the tree.

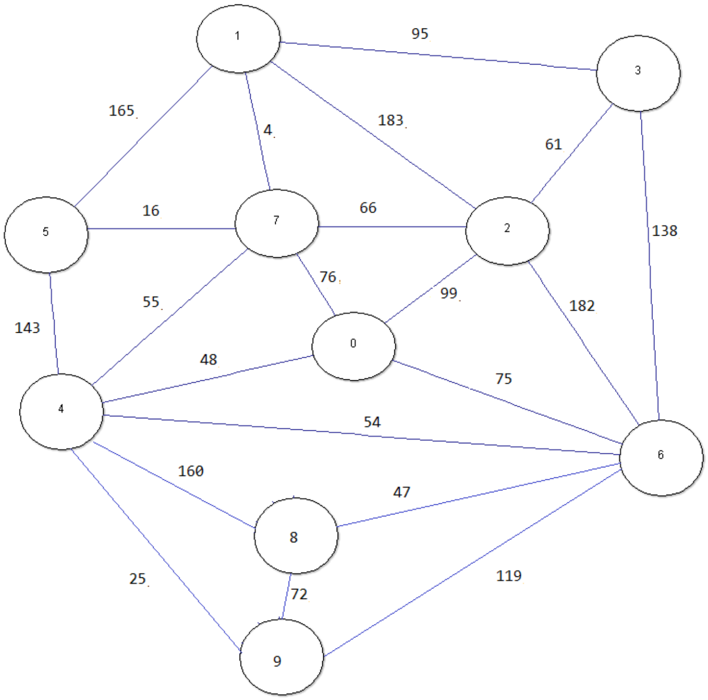


6 (10 Points) Insert the following key value pairs, (key, value), `into an initially empty symbol table with initial size 13`, `(44,0) (65,1) (66,2) (95,3) (89,4) (33,5) (84,6) (76,7) (66,8) (52,9).` Provide the final contents of the symbol table using hashing with linear probing without resizing.

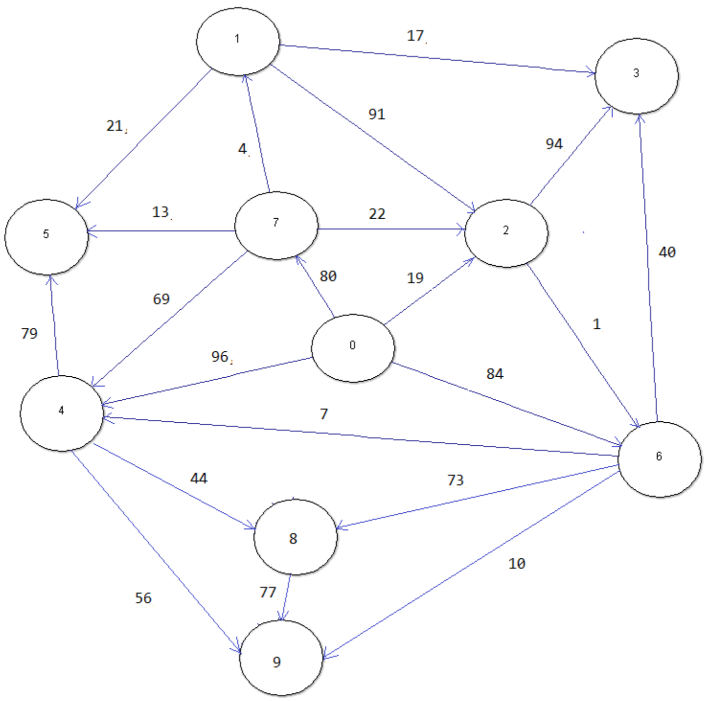
7. (10 Points) Given the following adjacency lists representation of a DAG: 1) How do you find a topological order?, 2) Provide a topological order.

```
15 vertices, 28 edges
0: 7
1: 5 9
2: 0 5
3: 0
4: 3 7 6 2
5: 8
6: 7 0 5
7:
8: 12 3
9: 11 12
10: 9 6 2
11:
12:
13: 2 8
14: 8 11 7 13 5
```

8. (10 Points) Use Lazy Prim's algorithm to find a Minimum Spanning Tree (MST) starting from vertex 0. Provide MST weight and indicate in which order the edges are added to MST



9. (10 Points) Given the edge-weighted digraph below,



1) Use Dijkstra’s algorithm to find the Shortest Path Tree (SPT) for source vertex 0. List the edges in the order as they are added to the SPT.
2) Provide the shortest distance from source vertex to every other vertexes.

10. (10 Points) Implement **isTopologicalOrder** method that checks whether or not a given permutation of a DAG's vertices is a topological order of that DAG. 1) Describe your algorithm, 2) Provide implementation in Java, 3) Provide worst-case space and time complexity in big-O notation.

```java
public class Digraph {
    private int V = 0; // number of vertices
    private int E; // number of edges
    private List<Integer>[] adj; // adjacency lists

    @SuppressWarnings("unchecked")
    public Digraph(int V)  {
        this.V = V;
        this.E = 0;
        adj = new LinkedList[V]; // Create array of lists.
        for (int v = 0; v < V; v++) {
            adj[v] = new LinkedList<Integer>();
        }
    }

    public int V() { return V; }

    public int E() { return E; }

    public void addEdge(int v, int w) {
        adj[v].add(0, w);
        E++;
    }

    public Iterable<Integer> adj(int v) {
        return adj[v];
    }

    /**
     * @param vertices a permutation of DAG's vertices
     * @param dag Input directed acyclic graph
     * @return true if the given vertices represent a topological order of the
                DAG, otherwise false
     */
    public static boolean isTopologicalOrder(List<Integer> vertices,
                                             Digraph dag) {
        // provide implementation here
    }
}
```