COMP 2230_02

Assignment 10

Colton Isles Kaylee Crocker

THOMPSON RIVERS UNIVERSITY

COMP 2230 – Data Structures and Algorithm Analysis

Assignment #10: Sets, Maps and Graphs

Due Date: Section 01 Nov. 28th Section 02 Nov 29th, 2024

Chapter 22 Sets and Maps

1. What is a set?

   A set is a collection of elements that contain unique (no duplicates) values with a primary purpose of determining if an element exists in the collection.
2. What is a map?

   A map is a collection that creates a relationship between a key and a value with the primary purpose of providing an efficient way of finding a value given its key. A map contains unique values and keys with one key matching the one value per key, but multiple keys can be mapped to the same value.
3. What is the difference between a set and a map?

   The difference between a set and a map is the purpose of the collections, the map is made for finding values whereas the set is made for seeing if it exists.

4. Create a Set ADT (set class to hold integers using an array as the underlying data structure) that has the following methods:
   a. add(int)
   b. remove(Int)
   c. boolean contains(int)

```
5. package Ass10_2230;

   import java.util.Arrays;

   public class SetADT {
       private int[] set;
       private static final int DEFAULT_SIZE = 5;

       SetADT() {
           set = new int[DEFAULT_SIZE];
       }

       public void add(int element) {
           if (!contains(element)) {
               for (int i = 0; i <= set.length; i++){
                   if (i == set.length) {
                       expandCapacity();
                   }
                   if (set[i] == 0) {
                       set[i] = element;
                       break;
                   }
               }
```

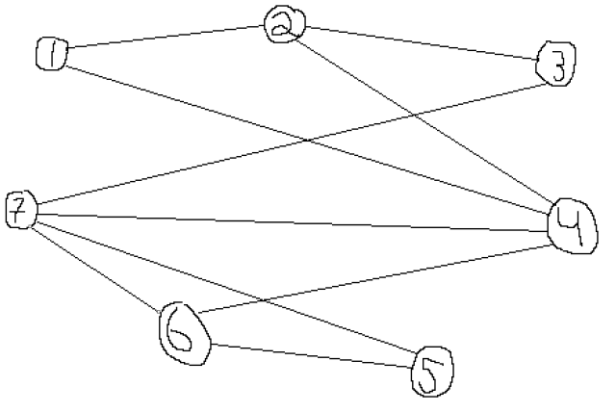
```
            }
        }

        private void expandCapacity() {
            set = Arrays.copyOf(set, set.length * 2);
        }

        public void remove(int element) {
            for (int i = 0; i < set.length; i++) {
                if (set[i] == element) {
                    set[i] = 0;
                    break;
                }
            }
        }

        public boolean contains(int element) {
            boolean result = false;
            for (int e : set) {
                if (e == element) {
                    result = true;
                    break;
                }
            }
            return result;
        }

        // just so we can see what's happening to test it
        public String toString() {
            return Arrays.toString(set);
        }
    }
```

Output:

```
[0, 0, 0, 0, 0]
[8, 5, 7, 12, 0]
[8, 5, 7, 12, 0]
[8, 0, 7, 12, 0]
[8, -26, 7, 12, 0]
```

Chapter 24 Graphs

1. Draw the undirected graph that is represented as follows:
   Vertices 1, 2, 3, 4, 5, 6, 7
   Edges (1, 2), (1, 4), (2, 3), (2, 4), (3, 7), (4, 7), (4, 6), (5, 6), (5, 7), (6, 7)



2. Is the graph from question 1 connected?
   Yes. The graph is connected as all vertices have a path to each other.
3. List all the cycles in the graph from question 1.
   Cycles (*Technically these are all the circuits):
   1→2→4→1
   1→2→3→7→4→1
   1→2→3→7→6→4→1
   1→2→3→7→5→6→4→1

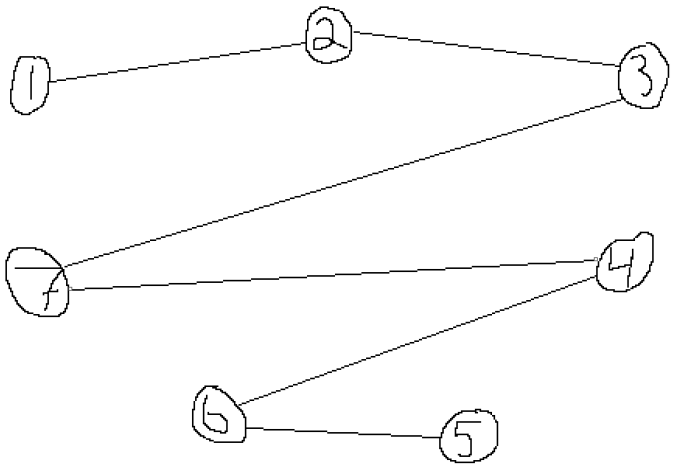
   2→3→7→4→2
   2→3→7→6→4→2
   2→3→7→5→6→4→2

   4→6→7→4
   4→6→5→7→4

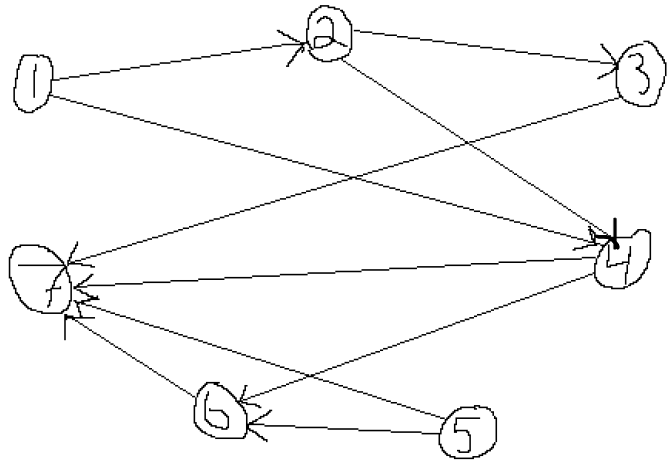
   5→6→7→5

   *All the cycles could be rewritten starting from any vertex in the cycle (Eg. 1→2→4→1 is also 2→4→1→2).
   *All the cycles could be rewritten backward (Eg. 1→2→4→1 is also 1→4→2→1).
   *Some of the cycles can be combined to create more larger cycles (Eg. Combine 1→2→4→1 with 4→6→7→4 to get the cycle 1→2→4→6→7→4 →1 that goes through 4 twice).

4. Draw a spanning tree for the graph from question 1.

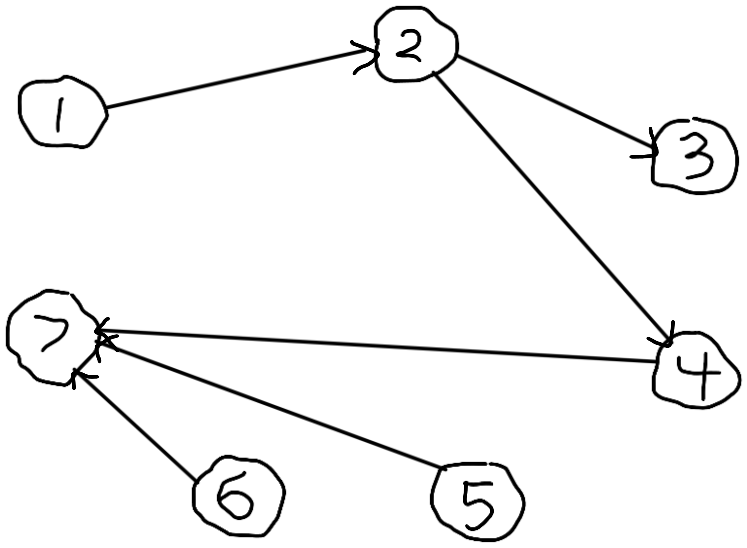


5. Using question 1, draw the resulting directed graph.



6. Is the directed graph of question 5:
    a. Connected?
    The directed graph is not connected as there is not necessarily a path from any vertex to any other (Eg. no path from 7 to 5).
    b. Complete?
    Graph is incomplete since there is not the maximum edges connecting vertices such as no edge between 4 and 5 and no edge between 1 and 7 etc.

7. List all of the cycles in graph of question 5.
    There are no cycles in the graph as no vertices can be both start and end. For vertices 1 and 5, there are only edges going away, so once you leave them, you can never come back. The opposite is true for vertex 7; you can arrive but never leave. The only way to arrive at 2 is from 1 (which we already determined does not work), so it cannot be part of a cycle either. The same logic applies when you try to leave 6 or 3 because of 7. After that, only 4 remains, which is not enough to create a cycle.
8. Draw a spanning tree for the graph of question 5.


1
2
3
4
5
6
7