Name: _____

Section: _____

CST 370 – Practice Midterm 1

Question:	1	2	3	4	5	6	Total
Points:	11	10	10	25	25	19	100
Score:							

Changelog:

- 2/22/19: Updated lists in problem 1.

1. Jasmine has written some code that sorts a list of integers and outputs partially sorted lists while it's running. Given the ouput of the program, what sorting algorithm did she implement?

   (a) (3 points) Identify the sorting algorithm Jasmine's code has implemented.

```
Input:
[39, 4, 16, 14, 6, 23]
Output:
[4, 39, 16, 14, 6, 23]
[4, 6, 39, 16, 14, 23]
[4, 6, 14, 39, 16, 23]
[4, 6, 14, 16, 39, 23]
[4, 6, 14, 16, 23, 39]
```

   (b) (4 points) Jasmine changed her code. Identify the new sorting algorithm Jasmine's code has implemented.

```
Input:
[17, 64, 8, 23, 4, 92, 3, 6]
Output:
[64, 23, 17, 6, 4, 8, 3, 92]
[23, 6, 17, 3, 4, 8, 64, 92]
[17, 6, 8, 3, 4, 23, 64, 92]
[8, 6, 4, 3, 17, 23, 64, 92]
[6, 3, 4, 8, 17, 23, 64, 92]
[4, 3, 6, 8, 17, 23, 64, 92]
[3, 4, 6, 8, 17, 23, 64, 92]
[3, 4, 6, 8, 17, 23, 64, 92]
```

   (c) (4 points) Jasmine changed her code again. Idenfity the new sorting algorithm Jasmine's code has implemented

```
Input:
[17, 64, 8, 23, 4, 92, 3, 6]
Output:
[17, 3, 8, 6, 4, 23, 64, 92]
[6, 3, 4, 8, 17, 23, 64, 92]
[3, 4, 6, 8, 17, 23, 64, 92]
[3, 4, 6, 8, 17, 23, 64, 92]
[3, 4, 6, 8, 17, 23, 64, 92]
[3, 4, 6, 8, 17, 23, 64, 92]
```

2. (10 points) Write an algorithm to determine if a graph is undirected given an Adjacency Matrix representation of the graph.

3. (10 points) Given the root of a binary search tree and integer target, implement the lookup function in any language of your choice. Have it return true if the target is contained in the tree, and false if it is not. For example, in C++, your functions signature might look like:

```
Hode* lookup(Node *root, int target)
```

4. Quicksort

   (a) (10 points) Given the following code for QuickSort, please implement the partition function. The code is provided in C++ but you may implement partition in any language of your choice. Use the middle element as your pivot choice.

```
void quickSort(vector<int> &vec, int l, int r) {
    if (l <= r) {
        int pivot = partition(vec, l, r);
        quickSort(vec, l, pivot - 1);
        quickSort(vec, pivot + 1, r);
    }
}
```

(b) (6 points) For the following array, what would the array look like after each call to **your partition function**?

```
[9, 8, 15, 10, 20, 3, 18]
```

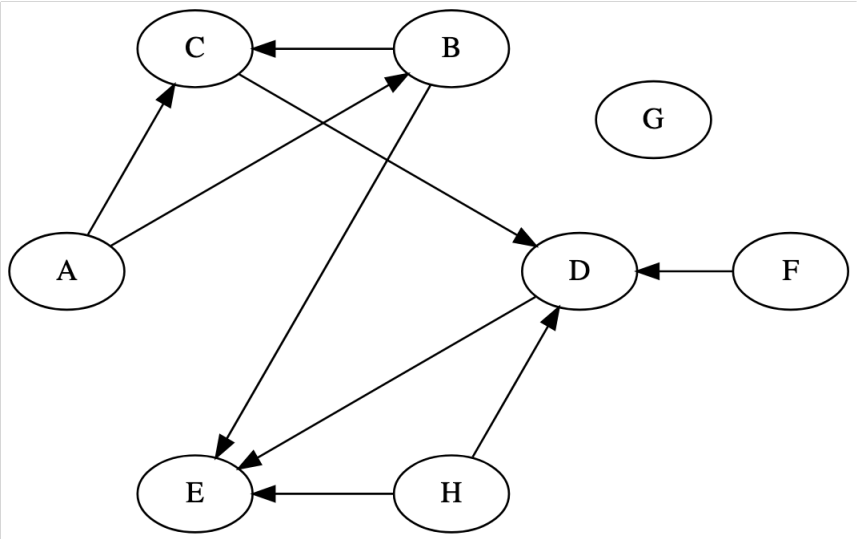
(c) (3 points) What are the best-, average-, and worst-case time complexities and the space complexity of QuickSort? For a recursive function like this, list Explicit Space complexity only. Remember to define your input size.

(d) (6 points) For both the best- and worst-case time complexities, justify in words why they are the same or different from the average-case complexity.

5. Galloping through Graphs

   (a) (10 points) Pick your favorite graph traversal algorithm (DFS or BFS) and implement it in any language of your choice to print the data in every node reachable from an origin node. You may choose how you wish to represent the graph: either adjacency list or adjacency matrix notation. You can assume each node holds data which is a single character.

(b) (6 points) Given the following graph, what is one possible DFS and one possible BFS, assuming origin A?



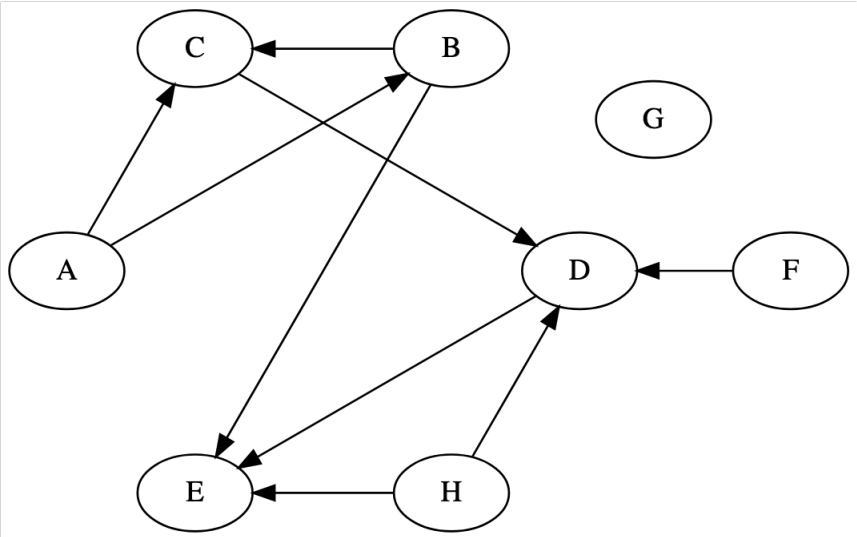
(c) (4 points) What are the best-, average-, and worst-case time complexities and the space complexity of your traversal algorithm? Why?

(d) (5 points) Will all the nodes in the graph necessarily be printed by a traversal starting at a single origin? Why or why not?

6. Topological Topics

   (a) (10 points) Pick your favorite topological sorting algorithm (modified DFS or Kahn's) and implement it in any language of your choice to print a topological sort of a graph. You may choose how you wish to receive the graph: either adjacency list or adjacency matrix notation.

(b) Given the following graph, what are **two** possible topological sorts based on input.



(c) (5 points) What is the purpose of topological sorting?

(d) (4 points) What subset of graph (connected versus unconnected, cyclic versus acyclic, directed versus undirected) can we perform topological sorts for? Justify why it is only this subset.