Instructions Follow instructions carefully, failure to do so may result in points being deducted. Clearly label each problem and submit the answers in order. Print out a copy of this cover sheet and staple it to the front of your assignment for grading. Be sure to show sufficient work to justify your answer(s). When asked to prove something, you must give a formal, rigorous, and complete proof. The CSE academic dishonesty policy is in effect (see https://cse.unl.edu/academic_integrity). You are highly encouraged to typeset your assignment using \LaTeX{}; if your answers are not legible, you may be required to use \LaTeX{} in future assignments.

Partner/Group Policy The same policy as your semester project is in effect.

Bonus Opportunity There are 100 points possible in this assignment. However, the assignment will be graded out of 50. Any points you earn beyond 50 will be treated as extra credit.

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Let $G = (V, E)$ be an undirected graph. The complementary graph, $\overline{G} = (V, \overline{E})$ has the same vertex set $V$, but its edge set is defined as:

$$\overline{E} = \{(u, v) \mid u, v \in V, (u, v) \notin E\}$$

That is, two vertices are adjacent in $\overline{G}$ if and only if they are not adjacent in $G$.

(a) Suppose that $|V| = n$ and $|E| = m$, what is the cardinality of $\overline{E}$?

(b) Consider graph $G_1$ in Figure 1. Draw $\overline{G_1}$

Let $G = (V, E)$ be a directed graph. The in-degree of a vertex $v$ is the number of edges incident into $v$.

(a) Design an algorithm (give pseudocode) that, given a vertex $v \in V$, computes the in-degree of $v$ under the assumption that $G$ is represented by an adjacency list. Give an analysis of your algorithm.

(b) Design an algorithm (give pseudocode) that, given a vertex $v \in V$, computes the in-degree of $v$ under the assumption that $G$ is represented by an adjacency matrix. Give an analysis of your algorithm.

A graph $G = (V, E)$ is bipartite if its vertices can be partitioned into two subsets $V = A \cup B$ such that all edges connect only vertices between the two sets (no two edges in the same set are connected).

(a) Prove or disprove: If a graph $G$ is bipartite, then $G$ is a tree.

(b) Prove or disprove: If a graph $G$ is a tree, then it is bipartite.

(Rosen 10.2.42) A sequence of integers $d_1, d_2, \ldots, d_n$ is called graphic if it is the degree sequence of a simple undirected graph. Determine whether each of these sequences is graphic. For those that are, draw a graph having the given sequence. For those that are not, provide some reason for why no graph has such a sequence.

(a) 5, 4, 3, 2, 1, 0
(b) 6, 5, 4, 3, 2, 1
(c) 2, 2, 2, 2, 2
(d) 3, 3, 3, 2, 2
(e) 3, 3, 2, 2, 2
(f) 1, 1, 1, 1, 1
(g) 5, 3, 3, 3, 3
(h) 5, 5, 4, 3, 2, 1

Let $G = (V, E)$ be an undirected graph.

(a) Prove or disprove: If $|E| \leq |V| - 1$ then $G$ is acyclic

(b) $G$ is said to be connected if every pair of vertices $u, v \in V$ is connected by some path. How big does the edge set $E$ have to be to guarantee that it is connected? (hint: its much larger than $n$).

Prove or disprove: A graph $G$ is always isomorphic to its complement graph $\overline{G}$; that is $G \cong \overline{G}$. 

Page 2
7. **Program: Graph Isomorphism Testing**

Two undirected graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ are said to be isomorphic, denoted $G_1 \cong G_2$ if there exists a bijective mapping $f : V_1 \rightarrow V_2$ such that for every pair of vertices $u, v \in V_1$, $(u, v) \in E_1 \iff (f(u), f(v)) \in E_2$.

An example of two isomorphic graphs can be found in Figure 1. These graphs are isomorphic as witnessed by the mapping $v_1 \rightarrow v_2, v_2 \rightarrow v_4, v_3 \rightarrow v_3, v_4 \rightarrow v_1$.

The *Graph Isomorphism Problem* is as follows: given two graphs $G_1, G_2$ determine whether or not $G_1 \cong G_2$. You will write a program to solve the graph isomorphism problem for any two graphs.

**Using JgraphT**

You will implement a static Java method, `getIsomorphicMap` in the Java source file provided on the course webpage. The method is detailed below.

```java
    /**
     * If the given graphs are isomorphic, returns a map between
     * the vertex sets witnessing the isomorphism. If the given
     * graphs are not isomorphic, returns null.
     */
    public static Map<String, String> getIsomorphicMap(
        UndirectedGraph<String, DefaultEdge> a,
        UndirectedGraph<String, DefaultEdge> b) {
        //TODO: implement
        return null;
    }
```

Note that this method uses a Java graph library called JGraphT ([http://www.jgrapht.org/](http://www.jgrapht.org/), jar files and a usage demo are available on the course webpage). To simplify things, the method has been defined to use only graphs whose vertices are represented by `String` types.

Hand in all of your source code via the webhandin program. Provide a short write-up of how your program works—what algorithm(s) did you use? What is the running time of your algorithm?