Problem 1. Complete the following implementation of the Bellman-Ford algorithm. `bellman_ford` should return whether the graph contains a negative-weight cycle.

```cpp
bool bellman_ford(vector<Node*> &V, Node* source, 
                  map<Node*, int> &D)
{
    D.clear();
    for (Node* v : V)
        D[v] = numeric_limits<int>::infinity();
    D[source] = 0;
    for (int i = 1; i < V.size(); ++i)
        for (Node* v : V)
            for (Node* nv : v->neighs)
                if (D[v] + v->weights[nv] < D[nv])
                    D[nv] = D[v] + v->weights[nv];
    for (Node* v : V)
        for (Node* nv : v->neighs)
            if (D[v] + v->weights[nv] < D[nv])
                return true;
    return false;
}
```
**Problem 2.** Determine the truth of the following statements about shortest-path algorithms.

If $|E| = \Theta(|V|)$, the worst-case running time of BFS is $O(|E|)$ □ True □ False

If $|E| = \Theta(|V|^2)$, the worst-case running time of Dijkstra’s is $\Theta(|V|^2 \log |V|)$ □ True □ False

Bellman-Ford is correct for disconnected input graphs. □ True □ False

For graphs with $|V| = \Theta(|E|)$, Dijkstra’s has lower asymptotic worst-case running time than Bellman-Ford. □ True □ False
Problem 3. Fill in the blanks with answers based on the graphs in Figure 2.

The number of edges of an MST is ______ and the weight of an MST is ______.

There are ______ distinct minimum spanning trees.

Removing the edge (_______, _______) increases the weight of an MST.

![Graph](image-url)

Figure 1: The graph for Problem 4.
Problem 4. Fill in the blanks with answers based on the graph $G$ in Figure 2.

The number of edges in any spanning tree of $G$ is ________.  

The weight of any MST of $G$ is ________.  

There are ________ distinct MSTs of $G$.  

Removing the edge (________,________) increases the weight of any MST of $G$.  

![Figure 2: The graph for Problem 4.](image)