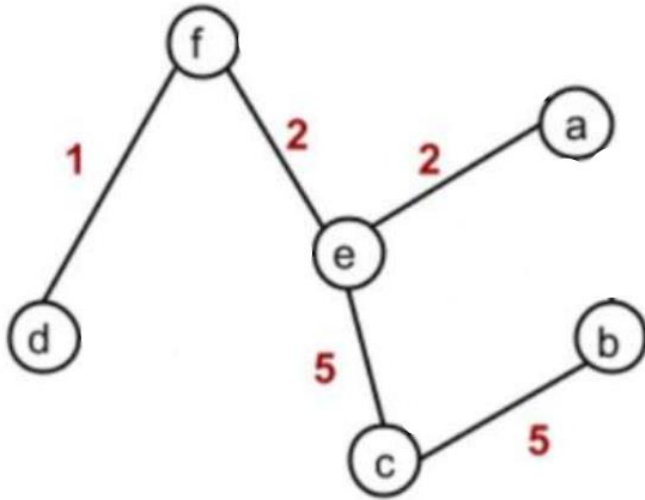


Variant 1

Q1) Solution (1.5 pts for the graph below, and 1 pt for weight = 15) :



Q2) Solution (2 pts for each highlighted line of code):

```
bool hasCycle(int node, int parent, vector<vector<int>>& adj,
vector<bool>& visited) {
    visited[node] = true;

    for (int neighbor : adj[node]) {
        if (!visited[neighbor]) {
            if (hasCycle(neighbor, node, adj, visited))
                return true;
        }
        // visited but not parent => cycle
        else if (neighbor != parent) {
            return true;
        }
    }
    return false;
}
```

Q3) Solution: $O(V+E)$

Variant 2

Q1) Solution (2 pts for each highlighted line of code):

```
bool hasCycle(int node, int parent, vector<vector<int>>& adj,
vector<bool>& visited) {
    visited[node] = true;

    for (int neighbor : adj[node]) {
        if (!visited[neighbor]) {
            if (hasCycle(neighbor, node, adj, visited))
                return true;
        }
        // visited but not parent => cycle
        else if (neighbor != parent) {
            return true;
        }
    }
    return false;
}
```

Q2) Solution: $O(V+E)$

Q3) Solution (1.5 pts for the graph below, and 1 pt for weight = 10):

