Quiz 3

Solution:

1) Main idea:

To detect whether the edge *e* (*u*,*v*) is in a cycle in graph, we simply take out this edge and run a BFS in this graph, if *u* is reachable from *v*, then that means there exist a cycle containing edge *e*; otherwise, edge *e* is not in the cycle.

2) Pseudocode:

```
Input: graph G, and edge e (u,v)
Output: True or False
Q is a queue
Q.enqueue(u)
while Q is not empty do:
    node = Q.dequeue()
    if node is v:
        return True
    for each node w connected to node v:
        if w is not labeled as visited then:
            label w as visited
            Q.enqueue(w)
    End
End
Return False
```

3) Proof of correctness:

If node *u* and its adjacent node *v* are in a cycle in an undirected graph, there must exist more than two paths from *u* to *v*. Therefore if we take out the edge (*u*,*v*), there must be more than one path connecting these two nodes. When we run the BFS, we start from *u* and search over all the nodes that connected to *u*. Then we can tell whether *v* is connected to *u* or not by checking if *v* is visited or not.

4) Time complexity O(V+E)