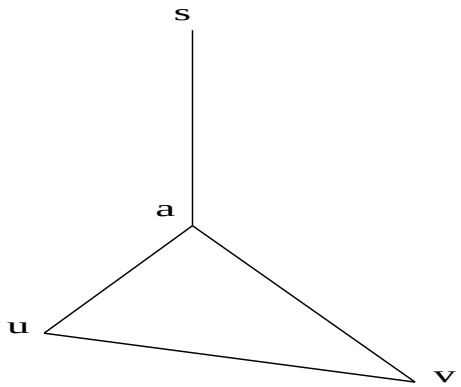
Homework 8

Problem 1 Solution: A modification of BFS is proposed. The algorithm is run on every connected component of the graph. Initially, an arbitrary vertex s is selected and the modified BFS is run. Actually, we enhance the existing code of BFS.

```
Modified-BFS(G,s) {

 Initialization /* exactly the same as in BFS, code is omitted for brevity */

While Queue is nonempty {
 u=Dequeue(Q);

for each v in Adj[u] {

 if (color[v]==white) {
      color[v]=gray;
      d[v]=d[u]+1;
      pred[v]=u;
      Enqueue(v);
   }

 else if (d[u] mod 2 == d[v] mod 2) {

        printf(``G has an odd cycle ``);
        return;
      }
}
color[u]=black;

}

}
```

Explanation: Look at the following figure for understanding why this works.



let $K(s,u)$, $K(s,v)$ represent the shortest path from s to u and v respectively, where u, v satisfy the above condition. This condition means that if both $d(u)$ and $d(v)$ are odd or even, then the path $K(s,u),(u,v)$, $K(v,s)$ forms an odd cycle (since adding 2 odd numbers or 2 even numbers gives even result. Adding 1 to an even number gives an odd number). Also, let a be the last common ancestor of the 2 paths. Then, the path $K(a,u),(u,v),K(v,a)$ also forms an odd cycle, since we actually subtract $K(s,a)$ 2 times from the original $K(s,u),(u,v)$, $K(v,s)$,which is odd, in order to form this path. $2K(s,a)$ is even, and subtracting an even number from an odd number gives an odd number.

The complexity of the algorithm remains the same as for BFS, since the extra condition costs constant time. That is, the complexity is $O(V+E)$.

Problem 2 Solution: Run DFS. As soon as the algorithm finds a grey vertex it exits. Note that it can not find a black vertex. Before finding a grey vertex, it sees only white vertex, i.e., it encounters a vertex at most once. Thus the complexity is O(V).

Problem 3 Solution: It does not always hold that v is a descendant of u. We provide the following counter-example:

Consider 3 nodes s, u, v with edges $(s,u),(s,v),(u,s)$. There exists a path from u to v: it's $(u,s),(s,v)$. If s is chosen as first vertex, and u is discovered first among its neighbors, then it holds that $d[u] < d[v]$. But v is not descendant of u in this case. QED

Problem 4 Solution: First, we prove that a back edge cannot exist. Assume that there exists a back edge (u, v) which connects u to one of its ancestors v in the tree. If v is not immediate ancestor, meaning that a path of length greater than 1 connects these 2 vertices, then u has another parent x in the tree. According to BFS, u is white when x is dequeued. But, this means that u was white even when v was dequeued. So, v would have color node u gray. This means that u was not white when x was dequeued, which is contradiction.

We use similar logic for the case of a forward edge (u, v). Let x be the parent of v in the BFS tree. v was white when x was dequeued. But v would have been colored gray when u was dequeued. So, u was not white when x was dequeued, which is contradiction.

Since there are not forward or backward edges, all edges in the graph are either tree or cross edges.