

Homework 4

Nikhil Unni (nunni2)

1. (a) The cut-vertices are f, g, h, j, and l.

(b)

```

low(G)
  Mark all nodes as having lowVal of -1
  time = 0
  while there is an unvisited node, u
    low(u)
    output lowVal, pre

low(u)
  pre(u) = ++time
  lowVal(u) = pre(u)
  for each edge (u,v) in Out(u)
    if lowVal(v) < 0 (unmarked)
      low(v)
      if lowVal(v) < lowVal(u)
        lowVal(u) = lowVal(v)
    else
      if pre(v) < lowVal(u)
        lowVal(u) = pre(v)

```

- (c) If we were to remove the root of the DFS tree, there are two cases : if there's a cross edge between the children trees of the root, or not.
 If there is no cross edge between any of the children, then it's obvious that if we were to remove the root, we would result in separate components for each of the children.
 If there is a cross edge between the children such that the removal of the root node would mean there's still only one component, then there must be a connection between all of the children. If this were the case, the DFS would have included all of the "children" as a single child in the search, which is guaranteed because it is an undirected graph. Thus, the only case this can happen is if the root only had a single child.
- (d) Because we know u is a non-root vertex, we know it has a parent, w . If u was a cut-vertex, then separating it would result in two distinct components, with no path possible between the two. And because the only way to connect the components is if one of u 's descendents had a connection to a proper ancestor of u (which we know has to exist!), if the two are truly to be two components, there is no way for any node in T_v to have a backedge to a proper ancestor of u . Similarly, if there were a connection between a descendent and ancestor of u , removing u would not result in two separate components, as they'd still be connected through that connection.

```

(e) cut(G)
    Mark all nodes as having lowVal of -1
    cutSet =  $\emptyset$ 
    time = 0
    while there is an unvisited node, u
        cut(u)
    output cutSet

    cut(u)
    pre(u) = ++time
    lowVal(u) = pre(u)
    if time = 1 and numEdges(u)  $\geq$  2
        cutSet = cutSet  $\cup$  {u}
    for each edge (u,v) in Out(u)
        if lowVal(v) < 0 (unmarked)
            cut(v)
            if lowVal(v) < lowVal(u)
                lowVal(u) = lowVal(v)
            if pre(u) > 2 and lowVal(v)  $\geq$  pre(u)
                cutSet = cutSet  $\cup$  {u}
        else
            if pre(v) < lowVal(u)
                lowVal(u) = pre(v)

```