University of Konstanz
Department of Computer & Information Science
Sabine Cornelsen / Julian Müller

Algorithms for Planar Graphs
Summer 2017

Assignment 3

Post Date: 15 May 2017 **Due Date:** 22 May 2017 **Tutorial:** 24 May 2017

Problem 1: DFS — 6 Points

Algorithm 1: Depth-First Search (DFS)

```
Input: undirected graph G = (V, E)
Data: counter d
Output: depth-first numbers DFS (initialized to ∞)
predecessor edges PARENT (initialized to NIL)
begin
    d ← 1;
    while there is r ∈ V with DFS[r] = ∞ do
        DFS[r] ← d++;
        Depth-First Search(r);

Depth-First Search(vertex v) begin
    while there is an unoriented edge {v, w} ∈ E do
        orient {v, w} from v to w;
        if DFS[w] = ∞ then
            DFS[w] ← d++;
            PARENT[w] ← (v, w);
            Depth-First Search(w);
```

Algorithm 1 shows a description of depth-first search. An edge is called *tree edge* if it is the parent of some vertex. All other edges are called *back edges*.

(a) Prove that the tree edges induce a spanning tree if G is connected.

(b) Let $\{v, w\}$ be a back edge oriented from v to w. Prove that w is on the unique r-v-path consisting of tree edges.

(c) Extend the depth-first search algorithm so that it also computes the lowpoints for all edges.

Problem 2: Outerplanarity **4 Points**

Show that a graph is *outerplanar* (has a planar drawing for which all vertices belong to the outer face) if and only if it does not contain a subdivision of K_4 or $K_{2,3}$.