ACM Summer School 2019

Graph Theory and Graph Algorithms

Problem sheet

Date: 18.06.2019 **Week:** 1 **Day:** 2

1. Show that the degree sequence is achieved by a tree if and only if the sum of degree is $2(n-1)$.

2. Prove that any two leaves of a DFS tree are not adjacent in the graph. Argue that for every edge in the graph (not necessarily in the tree) at least one of the endpoints is an internal vertex of the DFS tree.

3. Given an undirected graph G output an Eulerian traversal, if one exists. Recall that an Eulerian traversal is a sequence $v_1, e_1, v_2, \ldots, v_m, e_m, v_{m+1} = v_1$ where each edge e_i occurs exactly once, and for each $1 \leq i \leq m$, $e_i = \{v_i, v_{i+1}\} \in E(G)$.

4. Design an efficient algorithm to output the length of a simple cycle with least number of edges in an input graph G.

5. You are given a binary tree $T = (V, E)$ (in adjacency list format), along with a designated root vertex $r \in V$. A vertex u is called an anscestor of v in a rooted tree, if the path from r to v in T passes through u.

   We wish to preprocess the tree so that queries of the form "is u an anscestor of v?" can be answered in constant time. The preprocessing itself should take linear time. Suggest an algorithm for the same.

6. Give an efficient algorithm which takes as input a directed graph $G = (V, E)$ and determines whether or not there is a vertex $s \in V$ from which all other vertices are reachable.

7. If a graph has a hamiltonian path, then show that for every vertex set S, the number of components in $G \setminus S$ is at most $|S| + 1$.

8. Prove that a $4 \times n$ chess board has no knights tour. **Hint: Can a knights tour be represented as a Hamiltonian cycle in an appropriate graph?**