ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

Exercise 4 — **Graph Theory Applications**
Date: March 14, 2013 — Spring 2013

Problem 1. Let x be the number of leaves in the tree and y the number of all other vertices. Let e be the number of edges, then $e = x + y - 1$. Also, $\sum_{v \in V} deg(v) \geq x + 3y$ since every non-leaf vertex has degree ≥ 3. Using these two observations we have:

$$2(x + y - 1) = 2e = \sum_{v \in V} deg(v) \geq x + 3y \geq x + 3y - 2$$

from which $y \leq x$ follows.

Problem 2. If G is not a tree then it contains a cycle. There is least one edge (u, v) of this cycle which is not in G'. Obviously, for these vertices $d_G(u, v) = 1$, while in G' they are not neighbors, thus $d_{G'}(u, v) > 1$. For the second part, we simply start a breadth first search algorithm from vertex r (see text book for details).

Problem 3. It is easy to check that a graph that contains k edge-disjoint spanning trees on the n nodes satisfies property 3. In order to satisfy property 2, the graph should consist of exactly k edge-disjoint spanning trees, therefore contain $k(n - 1)$ edges. One way to construct such a graph is by starting with the complete graph on n vertices, find a spanning tree and remove it. Then on the remaining graph, find another spanning tree and remove it. Continue like this to find the k desired trees. In other words, starting with the complete graph keep exactly $k(n - 1)$ edges corresponding to k edge-disjoint spanning trees, and delete all other edges. Of course, you have to make sure that the graph does not get disconnected due to the removal of the spanning tree edges at any point of time. We don't prove it here, but for any $k \leq \lfloor n/2 \rfloor$ it is always possible to find such spanning trees.

Problem 4. The "only if" direction is easy. Let $T = (V, E)$ be a tree with n vertices. Since a tree has exactly $n - 1$ edges $\sum_{i=1}^{n} d_i = 2e = 2(n - 1)$. For the "if" direction, we will give a construction which given a degree sequence satisfying the given condition will produce a tree with that degree sequence.

Suppose that $d_1 \leq d_2 \ldots \leq d_n$ is the given degree sequence. We proceed by induction on the length of the sequence. The base case for $n = 2$ is trivial. Assume that the claim holds for all degree sequences of length less than n. Now for a degree sequence of length n, since d_1 is the lowest degree, $d_1 = 1$ (why?). Also, since d_n is the highest degree, $d_n \geq 2$ (why?). Consider the degree sequence $d_2, d_3, \ldots, d_n - 1$. There are $n - 1$ numbers summing to $2(n - 2)$. By the induction hypothesis, we have a tree T' corresponding to it. Now construct a new tree T with n vertices by gluing a single vertex to the vertex of degree $d_n - 1$ in T'. This completes the proof.

Problem 5. A tree with 100 vertices has 99 edges. Let x be the number of nodes with degree 10. All other nodes have at least degree one, so for the sum of degrees we have

$$2 \cdot 99 \geq 10x + 100 - x = 100 + 9x.$$

From this $x \leq 10$. It remains to show that $x = 10$ is possible. From the previous problem we know that it is enough to find a degree distribution with all positive degrees for which $\sum_{i=1}^{n} d_i = 2(n-1)$. E.g. the following distribution works: 10 vertices with degree 10 and 82 vertices with degree 1 and 8 vertices with degree 2.

Problem 6.

1. Clearly, property 1 and 2 for matroids is true. For the exchange property, let A and B be two subsets of linearly independent vectors such that $|A| > |B|$. Assume to the contrary that there is no $x \in A$ such that $B \cup \{x\}$ is also independent. Thus for all $e_i \in A$, $e_i \in Span(B)$. However, since $|A| > |B|$, at least two of the vectors in A are dependent, leading to a contradiction.

2. Again, properties 1,2 are trivial to check. For the exchange property, we use the fact that if a graph with n vertices, m edges and c connected components contains no cycles, then $n = m + c$. Now if $|A| > |B|$ and both A and B are independent, then the graph with edge set A has fewer components then the graph with edge set B, so some edge e of A must join vertices in different components of B; then adding e to B creates no cycle.

3. If we look at the weighted version of the matroid (2) above, the basis corresponds to precisely the spanning trees of the graph. Then, the basis with the minimum weight corresponds to the minimum spanning tree. The equivalence between the greedy algorithm for minimum weight basis and Kruskal's algorithm follows easily.