Solution to Problem Set 5

Date: 20.03.2014 Not graded

Problem 1. Let $R = BB^T$ and let d_i denote the i-the diagonal entry of matrix D. We need to show $(R)_{i,i} = d_i$ for $1 \leq i \leq n$, and $(R)_{i,j} = (A)_{i,j}$ for $1 \leq i, j \leq n$ and $i \neq j$. Let's start by looking at the diagonal entries of R. We have

$$(R)_{i,i} = \sum_{k=1}^{m} (B)_{i,k}(B)_{i,k} = \sum_{k=1}^{m} (B)_{i,k}^2$$

Since $(B)_{i,k} = 1$, iff vertex i neighbors edge k, we have that $(B)_{i,k}^2 = 1$ iff vertex i neighbors edge k. Hence $\sum_{k=1}^{m} (B)_{i,k}^2$ counts the number of edges incident to vertex i, which is just d_i. Since $(A)_{i,i} = 0$ for $1 \leq i \leq n$ in simple graphs, we conclude

$$(R)_{i,i} = (D)_{i,i} + (A)_{i,i}.$$

We now move to entries $(R)_{i,j}$, with $i \neq j$. We have

$$(R)_{i,j} = \sum_{k=1}^{m} (B)_{i,k}(B)_{k,j}$$

Again by definition $(B)_{i,k} = 1$ iff vertex i neighbors edge k, and $(B)_{k,j}^T = 1$ iff edge k neighbors vertex j. Therefore $(B)_{i,k}(B)_{k,j}^T = 1$ iff edge k neighbors vertices i and j. Hence $\sum_{k=1}^{m} (B)_{i,k}(B)_{k,j}^T$ counts the number of edges that join vertices i and j, which by definition equals entry $(A)_{i,j}$ of the adjacency matrix (for simple graphs, this number can either be 0 or 1). The problem statement follows.

Problem 2. No. The sum of the degrees is odd.

Problem 3. 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 Problem 3 of the previous set 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 4. *Cut Property:* To see the first claim, let $S \subset V$ be subset and let e be the edge with the given property. Assume that e is not already contained in the MST. We will show that this leads to a contradiction.

Add this edge to the spanning tree, hence creating a unique cycle. We can now drop from this cycle exactly the second edge which connects S to the outside, creating again a spanning tree. If e has, as given by assumption, strictly smaller weight, then the newly created spanning tree has strictly smaller weight, leading to the promised contradiction.

Cycle Property: Again we proceed by contradiction. Assume a MST does contain this edge. Then remove this edge from the spanning tree, so that we now get two components, let the vertices of these two components be called S and U and note that $S \cup U = V$. Note that the edge e has one end in S

and the other end in U. Let C be the cycle containing edge e. We claim that then C must contain at least one more edge, call it e' which has one end in S and one end in U. By definition the edge e' has smaller weight. Hence by adding it to the the two components we get a new spanning tree, but this tree has strictly smaller weight.

Problem 5.

1) The commune of Lausanne has to solve a minimum weight spanning tree problem, which can be done with Kruskal's algorithm (as seen in class).

2) Rudiger has to solve a maximum weight spanning tree problem. Indeed, Marco will check that there are no cycles, so Rudiger has to build a tree. In order to maximize the income of the telecommunications company, the tree needs to have the maximum possible weight. To solve the maximum spanning tree problem, Rudiger multiplies the edge weights by -1 and solves the minimum spanning tree problem on the new graph with Kruskal's algorithm.