MATH 249, WINTER 2017, ASSIGNMENT 3

This assignment is due Friday January 27 at 3pm. It should be submitted using crowdmark. The crowdmark instructions are the same as before.

Questions.

(1) **(5 points)** (Problems 1 and 2 from 5.3 of the MATH 239 course notes)

(a) Let r be a fixed vertex of a tree T. For each vertex v of T let $d(v)$ be the length of the path from v to r in T. Prove that
- for each edge $\{u, v\}$ of T, $d(u) \neq d(v)$, and
- For each vertex x of T other than r, there exists a unique vertex y such that y is adjacent to x and $d(y) < d(x)$.

(b) Let r be a fixed vertex in a graph G. Suppose that for each vertex v of G we have an integer $d(v)$ such that the two properties from the previous question hold. Prove that G is a tree.

(c) What does this question have to do with rooted trees?

(2) **(5 points)** Let G be a connected graph and suppose H is a spanning subgraph that has no cycles but any spanning subgraph of G that properly contains H does contain a cycle. Prove that H is a spanning tree.

(3) **(10 points)** Let G be a connected graph with $V(G) = \{v_1, v_2, \ldots, v_n\}$. Consider the following algorithm to build a subgraph H of G:
- Add v_1 to H.
- Repeat until $V(H) = V(G)$:
  - Find the vertex v that was added to H earliest and for which there is an edge in G from v to a vertex not in H.
  - Find the smallest j such that $\{v, v_j\}$ has one end in H and the other end not in H.
  - Add v_j, $\{v, v_j\}$ to H.

This algorithm builds a **breadth first seach tree** of G. It can be efficiently implemented using a queue.

(a) Prove the vertex v at the beginning of the loop will always exist provided $|V(H)| < |V(G)|$. What would happen if G were not connected?

(b) Prove that at the end of the algorithm H is a spanning tree of G.

(c) Say the root of a rooted tree is at **level** 0, and every other vertex is at **level** one more than its parent. Consider v_1 to be the root of H. Prove that when a vertex enters H in the algorithm, it never enters at a level lower than the highest level currently in H.

(d) Let H be the result after running the algorithm. Prove that every edge of G which is not in H joins two vertices that are at most one level apart in H. *This is the fundamental property of breadth first search trees.*

(e) Use the fundamental property of breadth first search trees to show that the length of a shortest path between two vertices u and v in a graph G is equal to the level of v in any breadth first search tree of G with u as the root.

(4) **(5 points)** Let G be a graph. Choose an orientation for each edge of G and build the **oriented incidence matrix** of G (with respect to this orientation) as for the usual incidence matrix except that the i, jth entry is 1 if edge j begins at vertex i and the entry is -1 if edge j ends at vertex i.

(a) Prove that the oriented incidence matrix for a connected graph has rank one less than the number of vertices.

(b) Prove that if E is the oriented incidence matrix then $EE^T = L$ where L is the graph Laplacian.

(5) **(3 points BONUS)** Give a different proof of the matrix tree theorem. Cite any sources you use. I wonder how many different proofs we can come up with between us all.