Tutorial 8: Graph Algorithms

1 Transporting Boxes

(This question was suggested by Trevor Brown.) A specified vertex W on a graph G represents a warehouse. Various vertices of the graph hold boxes. It is required to transport all the boxes to the warehouse by following edges of the graph G. The cost of transporting a box on vertex v to the warehouse W equals $2\times$ the number of edges on the path from v to W that is used (you have to walk from W to v, retrieve the box, and then carry the box back to W). It is possible to transport only one box at a time.

1. Assuming that there is at most one box at each vertex of G, determine the minimum cost to transport all the boxes to the warehouse. Give a detailed pseudocode description of your algorithm and briefly justify its correctness. Here the input would consist of the graph $G = (V, E)$ (represented using adjacency lists), the designated vertex W, and a subset of vertices $V_0 \subseteq V$ which represents the vertices that hold boxes.

2. Now suppose that there can be more than one box at a vertex. Suppose that $N[v]$ denotes the number of boxes at a vertex v, where $N[v] \geq 0$ is an integer for all vertices $v \in V$. Again, determine the minimum cost to transport all the boxes to the warehouse.

2 Back Edges and Directed Cycles in DFS

1. In a depth-first search of a directed graph, prove that any back edge vw is contained in a unique directed cycle C in which the remaining edges in C are tree edges.

2. Modify the basic depth-first search algorithm so that, whenever a back edge vw is encountered during the search, the edges in the above-mentioned directed cycle are printed out. This should be done while the depth-first search is taking place, not afterwards.