HW6 Solution, CS330 Discrete Structures, Spring 2015

Instructor: Professor Edward M. Reingold
TA: Taeho Jung

April 21, 2015

1 Algorithm for Euler circuit

Suppose the given directed graph does have a euler circuit.

We can slightly modify the DFS algorithm in the lecture note (available online) to find the Euler Circuit. In the original DFS, we mark each node whether we have visited, but in our new DFS algorithm 'DFS_EDGE', we mark the edges and examine whether we have visited the edge every time we find a new edge to traverse. Then, we run DFS_EDGE starting from any node until we need to backtrack. At this point, we have found a circuit

Algorithm 1 Euler Circuit Search

1: Pick any vertex in the graph to run DFS_EDGE until we need to backtrack. When we need to backtrack, we have found a circuit. Store the circuit as the CURRENT_CIRCUIT.
2: Pick the first vertex along the circuit that has an outgoing edge not visited. Run DFS_EDGE starting with that vertex until we need to backtrack. At this point, we have found another circuit.
3: Splice the newly found circuit with CURRENT_CIRCUIT.
4: Repeat 2,3 until all edges are visited.
5: Return CURRENT_CIRCUIT.

2 Tree from n-cube graph Q_n

2.1 BFS

Q_{n+1} is two copies of Q_n with the edges connecting the corresponding vertices in the two copies. Therefore, the BFS tree T_{n+1} from Q_{n+1} and T_n from Q_n has the following recurrence relationship.

Find two copies of T_n: T'_n, T''_n. Then, add an edge from the root of T'_n to the root of T''_n. Then, let the root of T''_n be a child of the root of T'_n, after which the merged tree is T_{n+1}. Obviously, T_0 is just a vertex.

Then, the BFS tree produced from Q_n can be described with the above recurrence relation.

2.2 DFS

Unlike BFS, DFS tree depends on how the algorithm chooses the vertex at each recursion. Depending on the choice, the tree can be a path (Hamiltonian path in the graph) or just a normal tree having several branches.

3 Minimum-weight edge & Minimum Spanning Tree

We prove this by contradiction. Suppose e is the minimum-weight edge in the connected weighted graph and assume that there exists a minimum spanning tree MST who does not contain e. Since MST is a tree, adding any extra edge to it will produce a simple cycle. Then, if we add e to MST, $MST \cup \{e\}$ is a graph having a cycle. Then, we can remove any other edge in the cycle to make it a tree, which is also a spanning tree. Since we added e and removed another edge whose weight is greater than e, the newly created spanning tree has a smaller total weight, which contradicts our first assumption that MST has the minimum total weight. Therefore, it is not possible that the minimum spanning tree does not contain e.

4 1-degree vertex & cycle

We again prove this statement by contradiction. Suppose a connected graph G does not have a 1-degree vertex and assume that the graph does not have a cycle. Since G does not have a cycle, it must be a tree. Every tree has at least one leaf node whose degree is 1, which contradicts our first condition that the graph does not have any vertex with degree 1. Therefore, any connected graph that has no vertex of degree 1 must contain a cycle.