CS 6505 - Homework 8

Caitlin Beecham

3

(a) We show that starting at an arbitrary vertex, j, we can traverse a sequence of edges and return to j such that we traverse each edge exactly once. We use induction. However, first we need the following lemma.
Every non-trivial graph in which every edge has even degree contains a cycle.
Proof: Consider a maximal path in G. Since, every vertex in G has even degree, in particular, the endpoints p_0 and p_k of this path have even degree, and since they are endpoints of some path they have degree at least 1, which means they have degree at least two. Consider one such endpoint, p_k. The fact that this path is maximal means that it cannot be extended along this endpoint which means that the other (other than p_{k-1}, well, there could be many) neighbor of this endpoint already lies on p. So it is some p_j with $j \notin \{k, k-1\}$. Then the portion of the path from p_j to p_k union the edge $p_k p_j$ forms a cycle and we are done.
Now, we apply induction on the number of edges. For our base case, take $m = 0$. Then the trivial circuit works. For the inductive step, we assume that the statement holds for graphs with fewer than m edges and we wish to show it holds for graphs with m edges. So, we take a graph with m edges and where every vertex has even degree. By the previous lemma, there exists a cycle in G. Delete the edge set of this cycle. We are left with some components each with fewer than m edges. So, by the inductive hypothesis, each has an eulerian circuit rooted at any vertex. Now, to construct an eulerian circuit for the whole graph, we start at our start vertex and there are two cases. If this vertex belongs to the cycle we deleted (and readded) we travel around the cycle detouring along the eulerian circuits which existed in each component as we hit them (when we hit the vertices in C that overlapped with a component of G-C). Otherwise if our start vertex belonged to one of the components of G-C, we start traversing the eulerian circuit we have for that component, D_i, until we hit a vertex of $C \cap D_i = d_i$, then we go around C detouring into each eulerian circuit as before until we return to d_i at which point we complete the eulerian circuit we started in D_i, giving an eulerian circuit for the entire graph.

(b) We make a modified graph in which every vertex has even degree. We know that in an even graph there is a tour traversing each edge exactly once. However, in graphs with some odd vertices, some paths between these odd vertices will be traversed more than once. We need to decide which paths between these odd vertices to use. We do the following. Look at all vertices of odd degree in G. Make a separate graph H consisting of just these odd vertices with every edge (a complete graph) in which the weight on the edge uv is the length of the shortest path from u to v in G. Find a minimum weight perfect matching of H. Then, go back to G. Create G' by adding the edges in the matching M of H to G (with their respective weights). G' has an eulerian tour. Take this tour of G' and create a tour of G by traversing the shortest path between u and v in G whenever the edge uv (an edge from M which was originally not present in G) is traversed. This will be our desired tour, L. To start we know that every edge is visited at least once as this tour visited every edge in G' which includes all edges of G. We also know that this tour is of minimum weight. How? Assume not and there is some shorter tour T. Then, once again we know that every edge of G is visited in T. Look at the set of edges, F, that are visited more than once. These edges pairwise connect pairs of odd vertices in G via paths. Also, the degree of each odd vertex in this graph F is 1. (Otherwise we have visited an edge multiple times that didn't need to be visited multiple times). This set of edges must apparently have sum of weights strictly less than our tour L, which is impossible as we found a minimum weight perfect matching in H and this tour T would indicate that there is a matching of lower weight.