CME305 Sample Midterm I

1. **Matchings and Independent Sets**

   Assume that you are given graph $G(V, E)$, a matching M in G and independent S in G. Show that

   $$|M| + |S| \leq |V| .$$

   Solution: Let M^* be a maximum matching and S^* be a maximum independent set. For each of the $|M^*|$ matched pairs, S^* can include at most one of the vertex, and therefore $|S^*| \leq |V| - |M^*|$, or equivalently $|M^*| + |S^*| \leq |V|$. Since M^* and S^* are maximum, we know $|M| + |S| \leq |V|$ holds for all M and S.

2. **Unique Minimum s-t Cut**

   Given a network $G(V, E, s, t)$, give a polynomial time algorithm to determine whether G has a unique minimum s-t cut.

   Solution: First compute a minimum s-t cut C, and define its volume by $|C|$. Let $e_1, e_2, \ldots, e_k$ be the edges in C. For each e_i, try increasing the capacity of e_i by 1 and compute a minimum cut in the new graph. Let the new minimum cut be C_i, and denote its volume (in the new graph) as $|C_i|$. If $|C| = |C_i|$ for some i, then clearly C_i is also a minimum cut in the original graph and $C \neq C_i$, so the minimum cut is not unique. Conversely, if there is a different minimum cut C' in the original graph, there will be some $e_i \in C$ that is not in C', so increasing the capacity of that edge will not change the volume of C', thus $|C| = |C_i|$. In conclusion, the graph has a unique minimum cut if and only if $|C| < |C_i|$ for all i. The algorithm takes at most $m + 1$ computing of minimum cuts, and therefore runs in polynomial time.

3. **Chinese Postman Problem**

Imagine that you are a postman. You park your truck in your district, and you want to walk around delivering mail to every street in the district and then return to your truck. Also, you are efficient so you want to minimize the total number of streets that you have to visit.

This can be formulated as a graph problem: given a connected graph $G(V, E)$, find a closed walk of minimum length that traverses every edge at least once.

(a) Give a polynomial time algorithm that gives a closed walk of length at most $2|E|$.

(b) (Harder) Give a polynomial time algorithm that gives a closed walk of length at most $|E| + |V| - 1$.

Solution: First notice that traversing an edge more than once is equivalent to adding multiple copies of this edge to the graph. Therefore we can pose the problem this way: what is the minimum number of edges we need to duplicate, such that the new graph is Eulerian.

(a) Double every edge and find an Eulerian circuit.

(b) First compute a spanning tree T. Then arbitrarily pair up the odd vertices. For each pair, there is a unique path connecting them in T, and we add one copy of each edge in the path. Clearly in the new graph every vertex has even degree, so the new graph is Eulerian. Notice that adding two copies of an edge does not change the parity of any vertex, so we never need to add more than one copy. Now since all added copies are edges in T, and no edge is added more than once, the total number of added edges is at most the number of edges in T, which is $|V| - 1$.