COL 702, Tutorial Sheet 4

1. **Graph theory**
   (i) Show that in any graph there are at least two vertices of the same degree.
   (ii) Given a degree sequence $d_1, D_2 \ldots d_n$ such that $\sum_i d_i = 2n - 2$, construct a tree whose vertices have the above degrees.
   (iii) Show that in a complete graph of six vertices where edges are colored red or blue, there is either a red or a blue triangle.

2. Given a directed acyclic graph, that has maximal path length k, design an efficient algorithm that partitions the vertices into $k+1$ sets such that there is no path between any pair of vertices in a set.

3. A directed graph is Eulerian if the in degree equals out degree for every vertex. Show that an Eulerian graph admits a tour where every edge is visited exactly once. Design an efficient (linear time) algorithm to find such a tour.

4. Find a maximum subgraph of $G = (V, E)$ that has degrees of each vertex is at least k.

5. Describe an efficient algorithm to find the *girth* of a given undirected graph. The *girth is defined as the length of the smallest cycle.*

6. Prove that subgraph returned by Dijkstra's algorithm (or Bellman Ford) is a directed tree rooted at source with $n - 1$ edges.

7. Given a directed acyclic graph, design a linear time algorithm for computing a SSSP in $O(|V| + |E|)$ time.

8. Let A be an $n \times n$ adjacency matrix of a directed graph $G = (V, E)$ with $A_{i,i} = 0$. We define a operation $B = A \oplus A$ as follows
   $$B_{i,j} = \min_{1 \le k \le n} \{a_{i,k} + a_{k,k}\}$$
   Note the similarity with normal matrix multiplication where we use $\times$ and $+$ instead of $+$ and min.
   (i) Prove that $B_{i,j}$ equals the shortest path of at most 2 edges between vertex i and vertex j.
   (ii) Prove that $B = A \oplus A \oplus \ldots \ell \ \ times \oplus A$ stores the shortest path with at most ℓ edges between i and j in $B_{i,j}$
   (iii) Design a fast algorithm to compute A^ℓ under the operation $\oplus$

9. Given a graph G with negative weights (no negative cycles), we want to transform it to another equivalent graph G' that preserves the shortest paths of G but doesn't contain any negative weights.
   (i) If we add to all edges a weight greater than the largest negative weight, will shortest paths be preserved ?
   (ii) Let $d(v)$ be equal to the shortest path distance to v from source vertex s. Suppose we add to every edge (u, v), the weight $d(u) - d(v)$, i.e. the new weight $w'(u, v) = w(u, v) + d(u) - d(v)$. Then show that

   (a) $w'(u, v) \ge 0$
   (b) Between all pairs of vertices x, y, for two distict paths P_1 and P_2, $w(P_1) \ge w(P_2)$ iff $w'(P_1) \ge w'(P_2)$.

10. Modify Bellman-Ford algorithm so that you can find the *second shortest path* from a source vertex to all other vertices. The *second shortest* path must differ from the shortest path by at least one edge but it can have the same weight as the shortest path. You should be able to rigorously justify the correctness of your algorithm.