1. Recall the definition of a bipartite graph. Let $G(V, E)$ be a graph and (A, B) be a partition of V. We say that G is bipartite if all edges in E have one end-point in A and the other in B. More precisely, for all $(u, v) \in E$ either $u \in A, v \in B$ or $u \in B, v \in A$.

   (a) Prove that a graph is bipartite if and only if it doesn't have an odd cycle.

   (b) A graph is called k-regular if all vertices have degree k. Prove that if a bipartite G is also k-regular with $k \geq 1$ then $|A| = |B|$.

2. **(Kleinberg & Tardos 13.1)** *3-Coloring* is a yes/no question, but we can phrase it as an optimization problem as follows.

   Suppose we are given a graph $G(V, E)$, and we want to color each node with one of three colors, even if we aren't necessarily able to give different colors to every pair of adjacent node. Rather, we say that an edge (u, v) is *satisfied* if the colors assigned to u and v are different.

   Consider a 3-coloring that maximizes the number of satisfied edges, and let c^* denote this number. Give a polynomial-time algorithm that produces a 3-coloring that satisfies at least $\frac{2}{3}c^*$ edges. If you want, your algorithm can be randomized; in this case, the *expected* number of edges it satisfies should be at least $\frac{2}{3}c^*$.

3. A tournament is a complete directed graph i.e. a directed graph which has exactly one edge between each pair of vertices. A Hamiltonian path is a path that traverses each vertex exactly once. A random tournament, is a tournament in which the direction of all edges is selected independently and uniformly at random.

   (a) What is the expected value of the number of Hamiltonian paths in a random tournament?

   (b) Use part (a) to show that for every n, there is a tournament with n players and at least

   $$\frac{n!}{2^{n-1}}$$

   Hamiltonian paths.

4. Given an undirected graph $G = (V, E)$ with nonnegative edge costs satisfying the metric inequality and whose vertices are partitioned into two sets $V = R \cup S$, *Required* and *Steiner*, find a minimum cost tree in G that contains all the vertices in R and any subset of the vertices in S. Finding an optimal solution to this problem is NP-hard. Find and prove a factor 2 approximation algorithm for this problem.

5. Show that the Steiner Tree problem (see above) is NP-hard by reducing minimum vertex cover to the Steiner Tree problem.