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.
   Solution:
   $\Rightarrow$: Suppose that $G(V = A \cup B, E)$ is bipartite. Assume for contradiction that there exists a cycle $v_1, v_2, ..., v_k, v_1$ in G with k odd. Without loss of generality we may additionally assume that $v_1 \in A$. Using the fact that G is bipartite, a simple induction argument suffices to show that $v_i \in A$ for i odd and $v_i \in B$ for i even. But then $\{v_k, v_1\} \in E$ is an edge with both endpoints in A, which contradicts the fact that G is bipartite. Therefore a bipartite graph G has no odd cycles.
   $\Rightarrow$: Let G be a graph with no odd cycles. We will consider the case that G is connected; this is sufficient since if we can show that each connected component of a graph is bipartite, then it follows that the graph as a whole is bipartite. Let $d(u, v)$ denote the length of the shortest path between two vertices in G. Pick an arbitary vertex $u \in V$ and define $A = \{u\} \cup \{w \mid d(u, w) \text{ is even}\}$. Define $B = V \setminus A$. We claim that the partition $V = A \cup B$ demonstrates that G is bipartite. Assume for contradiction that there exists an edge $\{w, v\} \in E$ with $w, v \in A$ (or B). Then by construction $d(u, w)$ and $d(u, v)$ are both even (or odd). Let P_{uw} and P_{vu} be the shortest paths connecting u to w, and v to u respectively. Then the cycle given by $P_{uw}, \{w, v\}, P_{v,u}$ has length $1 + d(u, w) + d(u, v)$, which is odd, a contradiction. Therefore no such edge $\{w, v\}$ may exist and G is bipartite.

   (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 $|A| = |B|$.
   Solution:
   Since each vertex of G has degree k, we have that $|A| = \frac{1}{k}\sum_{a\in A} \deg(a)$ and $|B| = \frac{1}{k}\sum_{b\in B} \deg(b)$. Now, every edge in G has exactly one endpoint in A and exactly one endpoint in B. Thus

   $$\sum_{a\in A} \deg(a) = \sum_{b\in B} \deg(b) = |E|$$

   and thus $|A| = |B| = |E|/k$.

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 $\frac{2}{3}c^*$.
   Solution:
   Consider the algorithm that picks a color for each vertex uniformly and independently at random. Then for each edge $e \in E$, the probability that e is satisfied is $2/3$. Define the indicator random variables $X_e = \mathbb{1}_{\{e \text{ is satisfied}\}}$ for $e \in E$; then we compute using linearity of expectation:

   $$\begin{aligned}\mathbf{E}[\text{number of satisfied edges}] &= \mathbf{E}\left[\sum_{e\in E} X_e\right] \\ &= \sum_{e\in E} \mathbf{E}[X_e] \\ &= \sum_{e\in E} \mathbf{P}[e \text{ is satisfied}] = \frac{2}{3}|E|.\end{aligned}$$

Now, noting that the optimal number of satisfied edges can be no more than the total number of edges, i.e. $c^* \leq |E|$, we have for our algorithm: $\mathbf{E}[\text{number of satisfied edges}] = \frac{2}{3}|E| \geq \frac{2}{3}c^*$.

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?

   Solution:
   Consider a random tournament $G(V = \{1, 2, ..., n\}, E)$ on n vertices. Let S_n denote the set of all permutations of $1, 2, ..., n$. For each permutation $\sigma \in S_n$, we may consider the event A_σ that σ gives a path in G, i.e. that the directed edge $(\sigma(i), \sigma(i+1)) \in E$ for each $1 \leq i \leq n-1$. We note that $\mathbf{P}(A_n) = (1/2)^{n-1}$, the probability that all of the aforementioned edges are directed in the desired direction. Then the number of Hamiltonian paths in G is given by $\sum_{\sigma \in S_n} \mathbb{1}_{\{A_\sigma\}}$, and by linearity of expectation:

$$\mathbf{E}\left[\sum_{\sigma \in S_n} \mathbb{1}_{\{A_\sigma\}}\right] = \sum_{\sigma \in S_n} \mathbf{E}[\mathbb{1}_{\{A_\sigma\}}] = n!\mathbf{P}(A_\sigma) = \frac{n!}{2^{n-1}}$$

   (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.

   Solution:
   Since the expected number of Hamiltonian paths is equal to $n!/2^{n-1}$, there must exist a tournament on n players with $\geq n!/2^{n-1}$ Hamiltonian paths. If this were not the case, i.e. every tournament had strictly $< n!/2^{n-1}$ Hamiltonian paths, then the expectation (which, in the absence of a formal probability class can be thought of as an "average") would be strictly $< n!/2^{n-1}$.

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.

   Solution:
   Consider the algorithm that returns the minimum cost spanning tree, e.g. through Kruskal's algorithm, connecting the vertices of R and completely ignoring the optional nodes in S. That is, we compute the minimum cost spanning tree T on the subgraph of G induced by R.

   To prove the approximation factor, let T^* denote the optimal solution to the Steiner Tree problem. Consider doubling every edge of T^* to produce a graph where every vertex has even degree; thus there exists an Eulerian cycle C_E on this graph. The cost of this cycle is $c(C_E) = 2c(T^*)$. Start at an arbitrary point on C_E and follow the cycle around, keeping track of the order in which you first see each vertex in R. Since T^* contains every vertex of R, this ordering will look like $r_1, r_2, ..., r_{|R|}$ where every vertex of R shows up exactly once in the list. Now consider the path $P = r_1, r_2, ..., r_{|R|}$ in G. Using a shortcutting argument similar to the proofs of our TSP approximation algorithms, we note that $c(P) \leq c(C_E) = 2c(T^*)$ (as a reminder, the argument boils down to the fact that skipping from r_i to r_{i+1} in C_E, omitting any intermediate vertices, can only decrease the length of P compared to C_E by the metric inequality). Now, P is a specific example of a spanning tree connecting the vertices of R. Our algorithm returns the minimum cost spanning tree, so $c(T) \leq c(P) \leq 2c(T^*)$ as desired.

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

   (Not Actually a) Solution:
   Consider an instance $G(V, E)$ of the (unweighted, undirected, but still NP-complete!) minimum vertex cover problem. We define a new graph $H(W, F)$ in which to find a Steiner tree. Let $W = V \cup E$ and

   $$F = \{\{u, v\} \mid u, v \in V\} \cup \{\{v, e\} \mid v \in e \in E\}.$$

   That is, to form H we take the complete graph on V, add a vertex for every edge $e \in E$, and connect each such e to its two endpoints in V. Let $c(e) = 1$ for all $e \in E$ and take $R = E \subset W$ and $S = V \subset W$. (In order to more properly frame this in the context of Problem 4's formulation, we may also imagine adding all unspecified edges $\{u, v\}$ to the graph with cost equal to $d(u, v)$; this is called the graph metric on H.)

   I leave it as an exercise to the reader to prove why a Steiner tree corresponds to a vertex cover in this construction, and thus a minimum Steiner tree corresponds to a minimum vertex cover.