Problem 1 (4.20 in [BT])
Consider the following linear program and its dual

$$\begin{aligned} \max \quad & c^{\mathrm{T}}x \\ \text{subject to} \quad & Ax \le b \\ & x \in \mathbb{R}^n \end{aligned} \qquad\qquad \begin{aligned} \min \quad & b^{\mathrm{T}}y \\ \text{subject to} \quad & A^{\mathrm{T}}y = c \\ & y \ge 0 \\ & y \in \mathbb{R}^m \end{aligned}$$

Assume that both problems have an optimal solution.
We have proved complementary slackness within the lectures, *i.e.*, for any optimal solution pair (x^*, y^*) we have for each $j = 1, \dots, m$ that $y_j^* > 0$ implies $a_j^{\mathrm{T}} x^* = b_j$ (where a_j is the jth row of A). We will show that there exist an optimal solution pair (x^*, y^*) such that for each $j = 1, \dots, m$ we have either $y_j^* > 0$ or $a_j^{\mathrm{T}} x^* < b_j$ (strict complementary slackness).

(a) Fix some j. Suppose that every optimal solution has $y_j = 0$. Show that there exists an optimal solution x such that $a_j^{\mathrm{T}} x < b_j$ by considering the following linear program and its dual

$$\begin{aligned} \min \quad & -y_j \\ \text{subject to} \quad & A^{\mathrm{T}}y = c \\ & y \ge 0 \\ & -b^{\mathrm{T}}y \ge -d \end{aligned}$$

where d is the optimal objective function value of the primal/dual program.

(b) Prove strict complementary slackness by using part (a) and taking the average of all the solutions obtained for each index j.

Problem 2
We consider the *2-matching* or sometimes also called *disjoint cycle cover* problem. Let $G = (V, E)$ be an undirected complete graph and $c : E \to \mathbb{R}$ be weights on the edges. A 2-matching is a collection of vertex disjoint cycles that covers all the vertices. Here, we allow that an edge can occur multiple times, *i.e.*, an edge being doubled is considered to be a cycle (to be precise, we require the degree of every vertex to be exactly two). The weight of a 2-matching is the sum of the weights of all the edges that constitute the 2-matching.
Show how a minimum weight 2-matching can be computed in polynomial time.

Hint: Use minimum perfect matching on a bipartite graph as a subroutine. Construct a bipartite graph that has all the vertices of G in each partition. Add an edge between u and v ($u \ne v$) in opposite partitions with weight $c(\{u, v\})$.