Exercise 1 (Let’s try random and derandomize). We consider the \( NP \)-complete Max-Cut problem: Given an undirected graph \( G = (V, E) \), find a cut \( C = (S, V \setminus S) \) of maximum size, where:

\[
\text{size}(C) = \#\{uv \in E : u \in S \text{ and } v \notin S\}.
\]

**Question 1.1** Prove that outputting a random cut is a randomized \( \frac{1}{2} \)-approximation for the Max-Cut problem. Which upper bound on \( \text{OPT} \) did you use? Exhibit a family of tight instances.

We now want to derandomize this algorithm by making the “best random choice” for each vertex. We consider each vertex, one after the other in an arbitrary order \( u_1, \ldots, u_n \), and place each vertex on the side of the cut which maximizes the expected number of edges in the cut given the choices already made and assuming the next choices are random.

**Question 1.2** Let \( A_i \) and \( B_i \) denotes the left and right sides of the cut \( C_i \) obtained after inserting \( u_1, \ldots, u_i \). Show that:

\[
\mathbb{E}[\text{size}(C_{i+1}) | A_i, B_i, u_{i+1} \in A_{i+1}] - \mathbb{E}[\text{size}(C_{i+1}) | A_i, B_i, u_{i+1} \in B_{i+1}] = \text{deg}(u_{i+1}, B_i) - \text{deg}(u_{i+1}, A_i),
\]

where \( \text{deg}(u, A) = \#\{v \in A : uv \in E\} \).

**Question 1.3** Conclude that the greedy algorithm that puts each vertex on the side to which it has the less connections is a deterministic \( \frac{1}{2} \)-approximation, as it is a derandomized version of the random cut algorithm (this derandomization scheme is called the conditional expectation method).

**Question 1.4** Give a direct analysis for the approximation ratio of this greedy algorithm.

Exercise 2 (Steiner tree). Given an undirected connected graph with non-negative edge weights \( G = (V, E, w) \) and a subset of required vertices \( R \subset V \), find a minimum weight subgraph of \( G \) that connects all the required vertices together. The non-required vertices in the obtained subgraph are called Steiner vertices. This problem is \( NP \)-complete.

**Question 2.1** Show that one can compute an optimal solution in polynomial time if the Steiner vertices of an optimal solution are given.

**Question 2.2** Show that if we have an \( \alpha \)-approximation algorithm for the metric case (i.e., when \( E = V^2 \) and \( w(uv) \leq w(uz) + w(zv) \) for all \( u, v, z \in V \)), then we have an \( \alpha \)-approximation algorithm for the general case.
Consider the following algorithm for the metric case: output a minimum spanning tree of the subgraph induced by the required vertices.

▶ **Question 2.3)** Show that it is a 2-approximation. Exhibit a family of tight instances (consider a graph whose edges weight 2 between two required vertices and 1 otherwise).

▶ **Question 2.4)** Explicit the corresponding 2-approximation for the general case.