1. Let \( G = (V, E) \) be an arbitrary graph with edge weights \( w : e \rightarrow \mathbb{R}_{\geq 0} \). Let \( \text{OPT}(G, w) \) denote the weight of a maximum cut in \( G \), and let \( \text{WLO}(G, w) \) denote the weight of the worst locally optimal cut in \( G \) with respect to the flip heuristic.

Prove that \( \text{WLO}(G, w) \geq \frac{1}{2} \cdot \text{OPT}(G, w) \).

2. The procedure in Exercise 1 might run for a long time. There is a much faster randomized algorithm: Choose a random subset \( C \subseteq V \) with \( \mathbb{P}(v \in C) = \mathbb{P}(v \notin C) = \frac{1}{2} \) independently for all \( v \in V \). Let \( C^\star \) be an optimal solution.

Show that
\[
\mathbb{E}(w(C)) \geq \frac{1}{2} \cdot w(C^\star).
\]

3. Derandomize the randomized algorithm from Exercise 2 by the method of conditional expectation: We decide for the vertices \( v_1, \ldots, v_n \) of \( V \) one by one if they should be included into \( C \). The decision for \( v_i \) is based on the conditional expectation given the previous decisions.

Make the description of this algorithm precise, prove that it always computes a cut of weight at least \( \frac{1}{2} w(C^\star) \), where \( C^\star \) is an optimal cut, and prove that it runs in polynomial time.