1 Multiway Cut (50 points)

Consider the following two permutations \( \pi_1 \) and \( \pi_2 \) of \( [k] \), where \( \pi_1(1) = 1, \pi_1(2) = 2, \ldots, \pi_1(k) = k \) and \( \pi_2(1) = k, \pi_2(2) = k - 1, \ldots, \pi_2(k) = 1 \).

(a) (25 points) Consider a modification of the 3/2 approximation for Multiway Cut from Lecture 17: instead of choosing \( \pi \) uniformly at random from all permutations of \( [k] \), we choose \( \pi = \pi_1 \) with probability \( 1/2 \) and choose \( \pi = \pi_2 \) with probability \( 1/2 \). Prove that this modified algorithm is still a 3/2-approximation for Multiway Cut.

(b) (25 points) Using the previous part, design a deterministic 3/2-approximation for Multiway Cut. As always, prove the approximation ratio and polynomial running time.

2 Multicut in Trees (50 points)

Consider the multicut problem in trees. In this problem, we are given a tree \( T = (V, E) \), \( k \) pairs \( (s_i, t_i) \) of vertices, and edge costs \( c : E \to \mathbb{R}^+ \). A feasible solution is a set \( F \subseteq E \) such that for all \( i \in [k] \), \( s_i \) and \( t_i \) are in different connected components of \( T - F \). The objective is to minimize the total edge cost \( \sum_{e \in F} c(e) \).

Let \( P_i \) be the unique path between \( s_i \) and \( t_i \) in \( T \). Then we can write an integer linear programming formulation of this problem:

\[
\begin{align*}
\min & \quad \sum_{e \in E} c(e)x_e \\
\text{subject to} & \quad \sum_{e \in P_i} x_e \geq 1 \quad \forall i \in [k] \\
& \quad x_e \in \{0, 1\} \quad \forall e \in E
\end{align*}
\]

(a) (25 points) Write the dual of the LP relaxation of the above ILP (note: we did this in class for multicut!)

Suppose that we root the tree at an arbitrary vertex \( r \). Let \( \text{depth}(v) \) be the number of edges on the path from \( v \) to \( r \). Let \( \text{lca}(s_i, t_i) \) be the vertex \( v \) on the path from \( s_i \) to \( t_i \) whose depth is minimum. Suppose that we use the primal-dual method to solve this problem, where the dual
variable that we increase in each iteration corresponds to the violated (primal) constraint that maximized $\text{depth}(\text{lca}(s_i, t_i))$. After all primal constraints are satisfied, we do a “reverse cleanup” stage like in Steiner Forest, where we look at the edges we added in reverse order and remove them if we can do so while still having a feasible solution.

(b) (25 points) Prove that this is a 2-approximation. Hint: consider a path $P_i$ where the dual variable is nonzero. How many edges in the final solution can be on the path from $s_i$ to $\text{lca}(s_i, t_i)$, and how many can be on the path from $t_i$ to $\text{lca}(s_i, t_i)$?