Problem 1. We say that a graph \( G = (V,E) \) is bipartite if we can partition its vertices into two sets \( P \) and \( Q \) such that each edge \( e \in E \) has one of its endpoints in \( P \) and another one in \( Q \). (In other words, we have that \( P \cap Q = \emptyset, P \cup Q = V, \) and \( E \subseteq P \times Q \).)

(a) Show that if a graph \( G \) is bipartite then its walk matrix \( W \) has one of its eigenvalues equal to \(-1\). 
(b) Show that if a graph \( G \) is connected and its walk matrix \( W \) has one of its eigenvalues equal to \(-1\) then \( G \) is bipartite.

Note: This, in particular, means that the only connected graphs on which the random walks do not mix are the bipartite ones.

Problem 2. Let \( n = 2^k - 1 \), for some \( k \geq 1 \), and let \( T_n \) be a full binary tree graph on \( n \) vertices. Show that \( \lambda_2(T_n) = \Omega(1/n) \).

Hint: It might be helpful to prove and use the following inequality (that is a slight generalization of the path inequality we have proved in the class). For any weights \( w_1, \ldots, w_{n-1} > 0 \),

\[
 w \cdot \left( \sum_{i=1}^{n-1} w_i \cdot L^{(i,i+1)} \right) \geq L^{(1,n)},
\]

where \( L^{(i,j)} \) is a Laplacian of a graph consisting of only one edge \((i,j)\) and \( w := \sum_{i=1}^{n-1} \frac{1}{w_i} \).

Note: In the class, we proved a weaker lowerbound of \( \lambda_2(T_n) = \Omega(1/n \log n) \).

Problem 3. Let \( n = 2^k \), for some \( k \geq 1 \), and let \( H_n \) be a graph on \( n \) vertices whose vertex set is \( \{0,1\}^k \) and two vertices are connected in it iff they differ at exactly one coordinate.

(a) Verify that for any \( a \in \{0,1\}^k \), the vector \( v^a \) with coordinates given by \( v^a_i := (-1)^{a_i} \) is an eigenvector of \( H_n \).

(b) What is the eigenvalue that \( v^a \) corresponds to?

(b) What is the conductance \( \Phi_{H_n} \) of this graph (as a function of \( n \))? (Provide an upper and lower bound that are as close to each other as you can make it.)

Problem 4 (Extra credit). Let \( \phi(x_1, \ldots, x_n) \) be a 2-CNF formula in \( n \) Boolean variables \( x_1, \ldots, x_n \), i.e., \( \phi(x_1, \ldots, x_n) = \bigwedge_{j=1}^{k} \psi_j(x_1, \ldots, x_n) \) is a conjunction of \( k \) clauses \( \psi_j(x_1, \ldots, x_n) \), where each of these clauses is an alternative of only two literals. (A literal is some variable \( x_i \) or its negation \( \neg x_i \)).

Assume that we know that \( \phi(x_1, \ldots, x_n) \) is satisfiable, i.e., that there exists (at least one) assignment \( \sigma^* \) of Boolean values to the variables, that makes all the clauses satisfied simultaneously. Consider the following randomized algorithm for finding a satisfying assignment for \( \phi(x_1, \ldots, x_n) \):

1. Start with an arbitrary assignment \( \sigma \).
2. As long as, \( \sigma \) does not satisfy the formula \( \phi(x_1, \ldots, x_n) \):
   Pick a clause that is not satisfied by \( \sigma \) and one of the two literals in it at random. Modify \( \sigma \) by flipping the value of the variable corresponding to the chosen literal.

Show that this algorithm terminates with satisfying assignment in expected \( O(n^2) \) iterations.

\(^1\)Note that coordinates of \( v^a \) are indexed by vertices of \( H_n \) and these are binary \( k \)-dimensional vectors. So, the inner product \( a \cdot i \) in the definition of \( i \)-th coordinate of the vector \( v^a \) is well-defined.