Instructions. This is a take home final. It will be due at the end of the scheduled final time which is Friday December 16 at 11:30AM. Please drop off our final at room 8109 GHC. If you have any questions please feel free to ask us. We will post all clarification on the web page.

Collaboration is not permitted.

Good luck and enjoy.

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>105</td>
<td></td>
</tr>
</tbody>
</table>

(25) 1. Packing Trees using Multiplicative Weights

In this problem we are given a unweighted graph $G = (V, E)$ and we would like to find the maximum number of edge disjoint spanning trees of $G$. Unfortunately this problem is NP-Hard. Instead we shall consider an easier problem of finding a maximum packing of fractional spanning trees (FST) in $G$.

We say that $T = (V, E', \alpha)$ is a fractional spanning tree (FST) of $G$ if $T = (V, E')$ is a spanning tree of $G$ and $0 < \alpha \leq 1$. We can think of $T$ as a spanning tree with edge weights $\alpha$.

Definition 1 Let $T_1, \ldots, T_k$ with fractions $\alpha_1, \ldots, \alpha_k$ be a set of FSTs. Let the congestion of an edge $e$ be the sum of weights of all FSTs containing the edge $e$. We say that the set of FSTs is uncongested if all edges have congestion at most 1. Furthermore, define a quality of such a set of FSTs as $\alpha_1 + \ldots + \alpha_k$. That is the objective is to maximize the quality.
Let $OPT$ be the maximum possible quality for a set of uncongested FSTs. Our task is to design a polynomial time algorithm that will find a set of FSTs quality at least $(1 - \epsilon)OPT$.

1. **Hint 1:** The weights for the multiplicative update will be a nonnegative cost assigned to each edge, say, $c = c_1, \ldots, c_m$
   Let $\Phi$ be the sum of the edge costs show there always exists a unit-weight spanning tree with sum of costs at most $\frac{\Phi}{OPT}$.

2. **Hint 2:** As in above, start with the costs 1 on each edge. Each time you find some spanning tree, increase the weight of each used edge by multiplying it with a constant factor. Repeat this process sufficient amount of times.
   Explain why $\log \Phi$ provides an upper bound on the number of times you used any particular edge.
   Find a way to discover a multitude of trees while keeping $\Phi$ relatively small.

3. **Hint 3:** When we find a multitude of such trees, we just scale them so that the maximum congestion is one. We want to argue that the sum of weights is still $(1 - \epsilon)OPT$.

2. **Loewner Inequalities**

1. Using the path embedding argument prove that for any spanning tree $T$ of a connected unweighted graph $G$,

   $$L_T \preceq L_G \preceq str_T(G)L_T$$

   where $str_T(G)$ is the stretch of $G$ in $T$.

2. Let $A$ and $B$ be symmetric positive definite $n$ by $n$ matrices.
   Show that $A \preceq \alpha B$ if and only if $\lambda_{\max}(B^{-1}A) \leq \alpha$

3. Let that $G = (V, E, w)$ and $H = (V, E', w')$ be two weighted graphs. Let define the stretch between of $G$ in $H$ as

   $$str_H(G) = \sum_{e \in E} w(e) \cdot ER_H(e)$$

   Using problem 2 prove that for any subgraph $H$ of a connected weighted graph $G$,

   $$L_H \preceq L_G \preceq str_H(G)L_H$$

   where $str_H(G)$ is the stretch of $G$ in $H$.

4. Let that $G = (V, E, w)$ and $H = (V', E', w')$ be two weighted graphs such that $W \subset V \cap V'$. 
Definition 2 Let $x$ vary over column vectors corresponding to the vertices in $W$, $y$ vary over column vectors corresponding to the vertices $V \setminus W$, and $y'$ vary over column vectors corresponding to the vertices $V' \setminus W$. We define the Lowener inequality $L_G \preceq_W L_H$ if for all $x$:

$$
\min_y \left( x \begin{bmatrix} y \\ y' \end{bmatrix}^T L_G \begin{bmatrix} x \\ y \end{bmatrix} \right) \leq \min_{y'} \left( x \begin{bmatrix} y' \\ y'' \end{bmatrix}^T L_H \begin{bmatrix} x \\ y' \end{bmatrix} \right)
$$

(a) Let $G = (V, E, w)$ be a weighted graph over the vertices $V$. Suppose that $A$ is the Schur complement of $L_G$ after pivoting out some single vertex/variable, say, $v_n$ from $L_G$. Show that $A$ is the Laplacian of a graph, say $H$ on the vertices $V_1, \ldots, V_{n-1}$.

(b) Furthermore, suppose we pivoted out from $G = (V, E, w)$ all the vertices $v \in W \subset V$. Denote the residual graph with $H = (V', E', w')$ where $V' = V \setminus W$. Show that:

$$
L_H \preceq_{V'} L_G \preceq_{V'} L_H
$$

3. $\lambda_2$ of Balanced Binary Tree

Let $T_n$ be a rooted balanced binary tree with $n$ vertices (i.e. the depth of the left and right subtrees of every node differ by 1 or less). Show that $\lambda_2$ of its Laplacian is $\Theta(1/n)$.

4. Laplacian Pseudoinverses

Let $G$ be a connected graph with positive weights on its edges and let $L_G^+$ be its pseudoinverse.

1. Prove that $(L_G^+)^{ii}_{ii} > 0$.

2. Prove or disprove: look at any row of $L_G^+$ in which there exists a unique element with maximum absolute value. In the row that element must lie on the diagonal of $L_G^+$.

3. Prove: let $M$ be the maximum absolute value of an element in $L_G^+$. Prove that there exists a diagonal element with absolute value $M$.

5. Linear Sized Sparsifier

In this problem we prove a step that helps us show that every Laplacian (graph) has linear-sized $1 + \epsilon$ sparsifiers. You can assume the following lemma is true.

Lemma 1 Given vectors $v_1, v_2, \ldots, v_m \in \mathbb{R}^n$ with

$$
\sum_{1 \leq i \leq m} v_i v_i^T = I_n
$$

then there exists scalars $s_i \geq 0$ with $|\{i : s_i \neq 0\}| \leq 4n$ so that

$$
I_n \preceq \sum_{1 \leq i \leq m} s_i v_i v_i^T \preceq 9I_n
$$

Page 3
With the help of the above lemma, prove that for every weighted graph $G$ there exists a subgraph $H \subseteq G$ with at most $4n$ edges such that $L_G \preceq L_H \preceq 9L_G$. 