The suggested deadline by which to finish this assignment is May 13th; since this assignment is ungraded, you don’t need to turn it in – just compare your solutions with the solutions I give.

1. In this problem, we prove that the Janson inequality parameter $\Delta$ is at most $O(\log N)$ in the Garg-Konjevod-Ravi Group Steiner Tree algorithm for trees, as claimed in class. Recall from class that we have a tree $T$ with root $r$ and $n$ nodes. There are $k$ disjoint groups $S_1, S_2, \ldots, S_k$, all of which are sets of leaves; also, $\max_i |S_i| = N$. Specifically, we fix a group $S_i$, and want to show that $\Delta_i$, the Janson inequality parameter for the set of leaves that correspond to $S_i$, is at most $\ln|S_i|$. 

**(a).** Suppose $j,j' \in S_i$. We will say that $j \sim j'$ if and only if (i) $j \neq j'$ and (ii) the least common ancestor of $j$ and $j'$ in $G$ is not the root $r$. If $j \sim j'$, let $lca(j,j')$ denote the least common ancestral edge of $j$ and $j'$ in $T'$. Show that

$$\Delta_i = \sum_{j,j' \in S_i : j \sim j', x_{lca(j,j')} > 0} \frac{x_{pe(j)}x_{pe(j')}}{x_{lca(j,j')}}.$$ 

**(b).** We will now prove the following key fact:

If $x_{pe(j)} > 0$, then $x_{pe(j)} \cdot \sum_{j' \in S_i : j \sim j'} \frac{x_{pe(j')}}{x_{lca(j,j')}} \leq x_{pe(j)} \ln(1/x_{pe(j)})$. \hfill (1)

We now take some steps toward proving (1). Suppose $x_{pe(j)} = z \in \langle 0, 1 \rangle$. We need some extra notation. Let $e_0, e_1, \ldots, e_\ell$ be the sequence of edges that we encounter as we walk up the tree starting from $j$; let $y_\ell = x_{e_\ell}$. Thus we have $z = y_0 \leq y_1 \leq y_2 \leq \cdots \leq y_\ell \leq 1$. Next, for $\ell = 0, 1, \ldots, \ell$, let $A_\ell = \sum_{j' \in (T(e_\ell) \cap S_i)} x_{pe(j')}$. Then, it is not hard to see that the left-hand side in the statement of (1) equals

$$z \cdot \sum_{\ell=1}^t A_\ell - A_{\ell-1} \cdot y_\ell. \hfill (2)$$

The sum in (2) is clearly bounded by the maximum of the following optimization problem, whose variables are the $y_\ell$ and $A_\ell$. (The optimization problem has a maximum since the domain is a polytope and since the objective function is continuous in the domain.)

$$OPT(z,t): \quad \text{maximize } \sum_{\ell=1}^t \frac{A_\ell - A_{\ell-1}}{y_\ell} \text{ subject to}$$

$$A_0 = z; \quad y_0 = z; \quad y_\ell \leq 1; \quad A_\ell \leq A_{\ell+1}, \quad \ell = 0, 1, \ldots, t-1; \quad y_\ell \leq y_{\ell+1}, \quad \ell = 0, 1, \ldots, t-1; \quad A_\ell \leq y_\ell, \quad \ell = 0, 1, \ldots, t. \hfill (3)$$

Constraint (3) holds since the following constraint (4) is a constraint in our LP relaxation:

$$\sum_{j \in (L(f) \cap S_i)} x_{pe(j)} \leq x_f \quad \text{for every edge } f \text{ and every group } S_i. \hfill (4)$$

Fix any feasible solution $\{y_\ell, A_\ell : \ell \geq 0\}$ to the above optimization problem.
• If \( v \) is the objective function value of this solution to the optimization problem, show that
\[
v \leq 1 - \frac{z}{y_1} + \sum_{\ell=1}^{t-1} \left(1 - \frac{y_\ell}{y_{\ell+1}}\right).
\]
\[(5)\]
• Take any \( \ell, 2 \leq \ell \leq t - 1 \). If we keep all variables but \( y_\ell \) fixed, see when the r.h.s. of (5) is maximized. Start with this idea to show that
\[
v \leq 1 - \frac{z}{y_1} + \ln\left(\frac{1}{y_1}\right).
\]
\[(6)\]
• Use (6) to show that \( v \leq \ln\left(\frac{1}{z}\right) \). This will then prove (1).

(c). Show, using (1), that \( \Delta_i \leq \ln|S_i| \).

2. We have a set \( V \) of \( n \) elements, and \( m \) distinct subsets \( S_1, S_2, \ldots, S_m \) of \( V \), each having cardinality \( t \). Our goal is to choose a subset \( W \) of \( V \) with “many” elements, subject to the constraint that no \( S_i \) (for \( i = 1, 2, \ldots, m \)) be a subset of \( W \).

Consider the following algorithm \( A \) for this problem. Let \( V \) be the set \( \{1, 2, \ldots, n\} \). Independently for each \( i \in V \), choose a number \( X_i \) uniformly at random from the set \( \{1, 2, \ldots, n^3\} \). Now define a set \( W \) as follows: for each \( i \in V \), \( i \in W \) iff there is no set \( S_j \) such that: (i) \( i \in S_j \), and (ii) for all \( k \in S_j \), \( X_i \geq X_k \).

(a). Show that \( A \) always produces a feasible solution to our problem.

(b). Suppose \( i \in V \) lies in \( a_i \) of the sets \( S_1, S_2, \ldots, S_m \). Show that the expected size of the set \( W \) produced by \( A \) is at least
\[
\left(\frac{1}{n^3}\right) \cdot \sum_{i=1}^{n} \sum_{\ell=1}^{n^3} \left(1 - \left(\frac{\ell}{n^3}\right)^{t-1}\right)^{a_i}.
\]

3. Suppose \( G \) is an undirected graph with maximum degree \( \Delta \). Each vertex \( u \) has a given color-list \( L(u) \), such that \( |L(u)| \geq \Delta + 1 \); we want a valid “list coloring” (i.e., an assignment of one color from \( L(u) \) to each \( u \), such that adjacent vertices get different colors.) Convince yourself that \( G \) has a valid list-coloring in this case; in this problem, we explore distributed list-coloring in the synchronous round-by-round model. (Thus, in a given round, any vertex communicates only with its neighbors.) Consider the following distributed algorithm for list-coloring. A generic round proceeds using the following four steps:

(S1) Each yet-uncolored vertex first wakes up with probability \( 1/2 \); if it chooses to not wake up in this round, it does not nothing for the rest of this round. (Thus, the three steps below only refer to vertices that woke up this round.)

(S2) \( u \) chooses a tentative color at random from its current list \( L(u) \).

(S3) Each vertex \( u \) that has some neighbor that chose the same tentative color as \( u \), is called unsuccessful; all other (yet-uncolored) vertices that woke up this round, are called successful.

(S4) Each successful vertex \( v \) is permanently given its chosen tentative color \( c \), and this color \( c \) is removed from \( L(w) \) for all neighbors \( w \) of \( v \) such that \( c \in L(w) \). The other vertices proceed to the next round.
Note that once a vertex gets a permanent color, it is never considered again.

(a) Show that step (S2) is well-defined: if \( u \) is yet-uncolored, show that \( L(u) \neq \emptyset \).

(b) Show that (if and) when the algorithm terminates, we have a valid list-coloring.

(c) Suppose we have some \( t \) yet-uncolored vertices at the beginning of a round. Show that the expected number of yet-uncolored vertices at the end of that round is at most \( 3t/4 \).