Problems

The examination will be graded blindly. Please do not write your name outside of this page; use your assigned code instead. Please use the answer sheets provided. Start each problem on a separate page and do not write outside the box. Anything outside the box may not be reproduced during scanning and will not be taken into account. Please write legibly in black ink on one side only. Your solution will be scanned and delivered electronically. Anything that interferes with scanning, such as bent or damaged pages, may delay grading. Please make sure that each page is marked with your student code, problem number and page number. Please number pages within each problem and for each problem please list the total number of pages submitted:

Problem	1	2	3	4	5	6
No. Pages						
Score						

I have worked on this examination on my own. I have neither sought nor received help from anyone else. I understand that making a false statement is a violation of the Georgia Tech Honor Code.

Signature: ____________________ Date: ______________

1. Computability, Complexity and Algorithms

Problem (dismantling a tree). You are given a weighted tree and your task is to dismantle it in the cheapest possible way. The only operation you are allowed is the following: you can erase any path such that all edges in the path have the same weight ω, and the price of doing this is ω. The tree is dismantled once you erased all its edges. Design an algorithm that returns the minimal cost to dismantle a given tree with weights.

Your input is a tree $T = (V, E)$ with vertex set $V := \{0, 1, \ldots, n-1\}$ and edge set $E := \{(u_1, v_1), (u_2, v_2), \ldots, (u_{n-1}, v_{n-1})\}$, where $u_i, v_i \in V$, for all i. You are also given the weight of each edge: let an integer $0 < w_i < K$ be the weight of edge (u_i, v_i). Your output should be a single integer denoting the minimal cost to dismantle the input tree. Explain why your algorithm is correct and analyze its running time in terms of n and K.

Solution. We start by making the following observation: if a node v has m adjacent edges with weight ω, then it costs $\omega(m/2)$ if m is even, and $\omega((m+1)/2)$ if m is odd. To simplify the notation, below we call this value $c(\omega, m)$ regardless of the parity of m. This observation follows from the fact that on each path you can only erase none or two such edges (because the graph is a tree). Observe that this cost is also tight: we cannot erase these edges cheaper than this.

We propose the following algorithm:

- Root the tree at vertex 0, and set $p(v)$ to be the parent of vertex $1 \leq v \leq n-1$ and -1 to be the parent of the root and $\omega_{-1} = 0$.
- For each vertex, set variables $\ell_v = (p(v), \omega_{p(v)}, \texttt{freq}[])$, where $\omega_{p(v)}$ is the weight of the edge connecting v to its parent, and $\texttt{freq}[]$ is a list of length K which stores the number of edges connected to v of weight i, at position i. We also set a variable `Cost` that we initialize to zero.
- Run DFS from the root. When visiting a vertex v for the first time, decrease $\texttt{freq}[\omega_{p(v)}]$ by one, to account for the edge connected to the parent that will be considered (i.e., erased) for some ancestor node. Then update `Cost` by adding $c(\omega, \texttt{freq}[\omega])$ for all $1 \leq \omega \leq K$ with $\omega \neq \omega_{p(v)}$, and update `Cost` with $c(\omega, \texttt{freq}[\omega]) - 1$, for $\omega = \omega_{p(v)}$.
- Return `Cost`.

As for the running time, there are $O(n)$ rounds of DFS, and each round populates an array of length K. The running time is $O(nK)$.

2. Theory of Linear Inequalities

Problem. Let $P = \{x \in \mathbf{R}^n : \sum_{i \in S} x_i - \sum_{i \notin S} x_i \leq 1\,,\ \ \forall S \subseteq \{1, 2, \ldots, n\}\}$. Show that

1. (4 pts) e_i and $-e_i$ are vertices of P, where e_i is the ith standard vector in $\mathbf{R}^n$.
2. (3 pts) All the 2^n inequalities, $\sum_{i \in S} x_i - \sum_{i \notin S} x_i \leq 1$ (for each subset S), are facets of P.
3. (3 pts) Show that $P = conv(\{e_1, e_2, \ldots, e_n, -e_1, -e_2, \ldots, -e_n\})$.

Solution.

1. Consider any e_i and constraints for sets $S_0 = \{i\}, S_1 = \{i, 1\}, S_2 = \{1, 2, i\}, \ldots, S_{i-1} = \{1, \ldots, i-1, i\}, S_{i+1} = \{1, \ldots, i-1, i, i+1\}, \ldots, S_n = \{1, \ldots, n\}$. Observe that e_i satisfies all these n constraints at equality. Moreover, the characteristic vectors of these constraints are linearly independent (Please verify). Thus e_i is a vertex. For $-e_i$, consider the complements of sets $S_0, \ldots, S_n$ as the corresponding n constraints.

2. Consider the constraint $\sum_{i \in S} x_i - \sum_{i \notin S} x_i \leq 1$ for some S. Then $F = \{x \in P : \sum_{i \in S} x_i - \sum_{i \notin S} x_i = 1\}$ contains the vertices $\{e_i : i \in S\} \cup \{-e_i : i \notin S\}$. Since these are n affinely independent vectors in F, we obtain dimension of F is at least $n - 1$. Since $F \neq P$, this implies that F must be a facet of P.

3. Clearly, $Q := conv(\{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}) \subseteq P$. We now show that $P \subseteq Q$. First observe that $0 \in Q$. Let $x \in P$. We claim that $\sum_{i=1}^n |x|_i \leq 1$ for all i. Let $S = \{i : x_i > 0\}$. Then we have $\sum_{i=1}^n |x_i| = \sum_{i \in S} x_i - \sum_{i \notin S} x_i$ which is at most 1 from the constraint in P. But then $x = \sum_{i=1}^n |x_i| sign(x_i) e_i + (1 - \sum_{i=1}^n x_i) 0$. Since $sign(x_i) e_i \in Q$ and $0 \in Q$, we obtain that x is a convex combination of points in Q and therefore is in Q proving the equality.

3. Graph Theory

Problem. Let $k \geq 3, t \geq 3$ be positive integers and let G be a graph with clique number k. Show that if G does not contain $K_{1,t}$ as an induced subgraph then $\chi(G) < R(k, t)$, where $\chi(G)$ as usual denotes the chromatic number of G and $R(k, t)$ denotes the Ramsey number with respect to clique of size k and independent set of size t.

Solution. First, we show $\Delta(G) < R(k, t)$. Let $v \in V(G)$ and consider $H := G[N(v)]$, the subgraph of G induced by the neighborhood $N(v)$ of v in G. Suppose $d(v) \geq R(k, t)$. Then H contains a clique $K \cong K_k$ or an independent set S of order t. If H contains K then $G[V(K) \cup \{v\}]$ is a clique of order $k + 1$, a contradiction. So H contains S. Now $G[S \cup \{v\}]$ is an induced $K_{1,t}$ in G, a contradiction.

Therefore, by Brooks' theorem, $\chi(G) \leq R(k, t)$. Now suppose $\chi(G) = R(k, t)$. Then, $\Delta(G) = R(k, t) - 1$ and G is a complete graph or G is an odd cycle. Note that $R(k, t) > k$ when $k \geq 3$ and $t \geq 3$. Thus G cannot be complete as the clique number of G is k. Moreover, $\Delta(G) = R(k, t) - 1 \geq 3$, so G cannot be an odd cycle. This contradiction implies $\chi(G) < R(k, t)$.

Problems

The examination will be graded blindly. Please do not write your name outside of this page; use your assigned code instead. Please use the answer sheets provided. Start each problem on a separate page and do not write outside the box. Anything outside the box may not be reproduced during scanning and will not be taken into account. Please write legibly in black ink on one side only. Your solution will be scanned and delivered electronically. Anything that interferes with scanning, such as bent or damaged pages, may delay grading. Please make sure that each page is marked with your student code, problem number and page number. Please number pages within each problem and for each problem please list the total number of pages submitted:

Problem	1	2	3	4	5	6
No. Pages						
Score						

I have worked on this examination on my own. I have neither sought nor received help from anyone else. I understand that making a false statement is a violation of the Georgia Tech Honor Code.

Signature: ________________________ Date: __________________

4. Analysis of Algorithms

Problem. Suppose that we have n unit squares $S_1, \ldots, S_n$ in the Euclidean plane, each of which has side length 1. They may not align with the axes and may overlap with each other. You are given the positions (coordinates of the four vertices) of all squares. Let $S = S_1 \cup \cdots \cup S_n$ be the *union* of these squares.

Part (a): Consider the following algorithm:
Pick i from $\{1, \ldots, n\}$ uniformly at random and then pick a point x from S_i uniformly at random. If $x \notin S_j$ for all $1 \le j < i$, then the algorithm succeeds and outputs x; otherwise it fails.
Prove that this algorithm succeeds with probability at least $1/n$, and when it succeeds the point it outputs is from the uniform distribution over S.

Part (b): Give a randomized algorithm that approximately estimates the area of S. More specifically, for given $0 < \varepsilon < 1$, the output Z of your algorithm should satisfy

$$\mathbf{Pr}\left[(1-\varepsilon)|S| \le Z \le (1+\varepsilon)|S|\right] \ge 3/4 \qquad (*)$$

where $|S|$ denotes the area of S. The running time of your algorithm should be polynomial in n and $1/\varepsilon$.

State your algorithm and its running time, and include the analysis of your algorithm. You may assume that for every square S_i, in $O(1)$ time you can check membership (is $x \in S_i$?), and in $O(1)$ time you can generate a point uniformly at random from S_i. You may neglect all numerical issues caused by real numbers.

Solution.

Part (a): The probability of success is

$$\begin{aligned}
&\sum_{t=1}^{n} \Pr(i=t) \cdot \Pr(\forall 1 \le j < t : x \notin S_j) \\
&= \sum_{t=1}^{n} \frac{1}{n} \cdot \frac{|S_t \backslash (S_1 \cup \cdots \cup S_{t-1})|}{|S_t|} \\
&= \frac{1}{n} \sum_{t=1}^{n} |S_t \backslash (S_1 \cup \cdots \cup S_{t-1})| \\
&= \frac{1}{n} |S_1 \cup \cdots \cup S_n| \\
&= \frac{|S|}{n} \\
&\ge \frac{1}{n}.
\end{aligned}$$

Let μ denote the distribution of the output of the algorithm when it succeeds. For every $x \in S$ let $i(x)$ be the smallest index i such that $x \in S_i$. Then

$$\mu(x) = \frac{\Pr(i = i(x)) \cdot \frac{1}{|S_i|}}{\Pr(\text{SUCCESS})} = \frac{1}{|S|}.$$

Part (b): The algorithm is as follows: in the k'th round, we pick i_k from $\{1, \ldots, n\}$ uniformly at random and then pick a point x_k from S_{i_k} uniformly at random (this can be done in constant time by our assumption). If $x_k \notin S_j$ for all $1 \leq j < i_k$, then let $Y_k = 1$; otherwise let $Y_k = 0$. The output of the algorithm is

$$Z = \frac{n}{N} \sum_{k=1}^{N} Y_k$$

where N is the number of rounds.

First we show that $\mathbb{E}[Y_k] = |S|/n$ for every k.

$$\begin{aligned}
\mathbb{E}[Y_k] &= \Pr(Y_k = 1) \\
&= \sum_{i=1}^{n} \Pr(i_k = i) \cdot \Pr(\forall 1 \leq j < i : x_k \notin S_j) \\
&= \sum_{i=1}^{n} \frac{1}{n} \cdot \frac{|S_i \backslash (S_1 \cup \cdots \cup S_{i-1})|}{|S_i|} \\
&= \frac{1}{n} \sum_{i=1}^{n} |S_i \backslash (S_1 \cup \cdots \cup S_{i-1})| \\
&= \frac{1}{n} |S_1 \cup \cdots \cup S_n| \\
&= \frac{|S|}{n}.
\end{aligned}$$

Note that $|S_1| = \cdots = |S_n| = 1$.

Then by the Chernoff bounds and the fact $|S| \geq |S_1| = 1$,

$$\begin{aligned}
\Pr\left(|Z - |S|| \geq \varepsilon |S|\right) &= \Pr\left(\left|\sum_{k=1}^{N} Y_k - \frac{N|S|}{n}\right| \geq \frac{\varepsilon N|S|}{n}\right) \\
&\leq 2\exp\left(-\frac{N|S|\varepsilon^2}{3n}\right) \\
&\leq 2\exp\left(-\frac{N\varepsilon^2}{3n}\right) \\
&\leq 1/4
\end{aligned}$$

when

$$N \geq \frac{cn}{\varepsilon^2}$$

for some constant c.

The running time is $O(nN)$.

An alternative algorithm is the following: in the k'th round, we pick i_k from $\{1, \ldots, n\}$ uniformly at random and then pick a point x_k from S_{i_k} uniformly at random. We then check how many squares S_j contain x_k and let

$$\alpha(x_k) = |\{j : x_k \in S_j\}|.$$

Let $Y_k = 1/\alpha(x_k)$. The output of the algorithm is

$$Y = \frac{n}{N} \sum_{k=1}^{N} Y_k$$

where N is the number of rounds.

We again show that $\mathbb{E}[Y_k] = |S|/n$ for every k.

$$\begin{aligned}
\mathbb{E}[Y_k] &= \sum_{i=1}^{n} \frac{1}{n} \sum_{x \in S_i} \frac{1}{\alpha(x)|S_i|} \\
&= \sum_{x \in S} \sum_{j : S_j \ni x} \frac{1}{n|S|} \frac{1}{\alpha(x)} \\
&= \frac{1}{n}.
\end{aligned}$$

The remainder of the analysis proceeds as before.

5. Combinatorial Optimization

Problem. Given a planar graph $G = (V, E)$ with weighted (multiple) edges, give a polynomial time algorithm to find the minimum-weight edge set M that can be removed such that the graph $H = (V, E \setminus M)$ is 2-colorable (i.e., every vertex can be colored with two colors, such that no two adjacent vertices have the same color).

Solution. We may assume that G is connected. Let $G^* = (V^*, E^*)$ be the dual graph. Then H is 2-colorable if and only if the graph $H^* = (V^*, E^*/M^*)$ obtained by contracting the edges dual to M is Eulerian. The latter condition is equivalent to M^* being a T^*-join in G^*, where T^* is the set of vertices of odd degree in G^*. Thus the problem is equivalent to finding a minimum weight T^*-join in G^*, which can be done by Theorem 29.1 in [Schrijver, Combinatorial Optimization].

6. Probabilistic methods

Problem. Given a finite set Γ, let Γ_p denote the random subset where each element $x \in \Gamma$ is included independently with probability p. Given any event $\mathcal{E}$ (a family of subsets of Γ), to avoid clutter we write $\Pr(\mathcal{E}) = \Pr(\Gamma_p \in \mathcal{E})$, as usual. Furthermore, we say that $\mathcal{E}$ is *increasing* if $\mathcal{X} \subseteq \mathcal{X}^+ \subseteq \Gamma$ and $\mathcal{X} \in \mathcal{E}$ imply $\mathcal{X}^+ \in \mathcal{E}$. Similarly, we say that $\mathcal{E}$ is *decreasing* if $\mathcal{X}^- \subseteq \mathcal{X}$ and $\mathcal{X} \in \mathcal{E}$ and imply $\mathcal{X}^- \in \mathcal{E}$.

Setup/What you may assume: Let $\mathcal{I}_1, \mathcal{I}_2, \dots$ denote increasing events, and let $\mathcal{D}$ denote a decreasing event. In the following sub-tasks you may only assume *Harris inequality* (a special case of the FKG inequality), which states that any two increasing events $\mathcal{I}$ and $\mathcal{J}$ are positively correlated, i.e., satisfy $\Pr(\mathcal{I} \cap \mathcal{J}) \geq \Pr(\mathcal{I}) \Pr(\mathcal{J})$.

(a) Show that $\Pr(\mathcal{I}_1 \cap \cdots \cap \mathcal{I}_k) \geq \Pr(\mathcal{I}_1) \cdots \Pr(\mathcal{I}_k)$ for all integers $k \geq 1$.
(b) Show that $\Pr(\mathcal{I}_1 | \mathcal{D}) \leq \Pr(\mathcal{I}_1)$.
(c) Show that if $\mathcal{I}_1$ and $\mathcal{I}_2$ are independent, then $\Pr(\mathcal{I}_1 \mid \mathcal{I}_2 \cap \mathcal{D}) \leq \Pr(\mathcal{I}_1)$.

Solution.
(a) We proceed by induction on k, where the base case $k = 1$ is trivial. For the induction step $k \geq 2$, note that the intersection of two increasing events is again an increasing event. Repeated application of this observation imply that $\mathcal{I}_1 \cap \cdots \cap \mathcal{I}_{k-1}$ is an increasing event. Using first Harris inequality and then the induction hypothesis, we thus obtain

$$\begin{aligned}\Pr(\mathcal{I}_1 \cap \cdots \cap \mathcal{I}_k) &= \Pr\big((\mathcal{I}_1 \cap \cdots \cap \mathcal{I}_{k-1}) \cap \mathcal{I}_k\big)\\ &\geq \Pr(\mathcal{I}_1 \cap \cdots \cap \mathcal{I}_{k-1}) \cdot \Pr(\mathcal{I}_k)\\ &\geq \Pr(\mathcal{I}_1) \cdots \Pr(\mathcal{I}_{k-1}) \cdot \Pr(\mathcal{I}_k),\end{aligned}$$

completing the proof.

(b) Noting that the complement of an increasing event is a decreasing event, we infer that $\mathcal{D}^c$ is an increasing event. Hence, after rewriting the desired probability, Harris inequality implies

$$\begin{aligned}\Pr(\mathcal{I}_1 \cap \mathcal{D}) &= \Pr(\mathcal{I}_1) - \Pr(\mathcal{I}_1 \cap \mathcal{D}^c)\\ &\leq \Pr(\mathcal{I}_1) - \Pr(\mathcal{I}_1) \cdot \Pr(\mathcal{D}^c)\\ &= \Pr(\mathcal{I}_1) \cdot \Big(1 - \Pr(\mathcal{D}^c)\Big) = \Pr(\mathcal{I}_1) \cdot \Pr(\mathcal{D}),\end{aligned}$$

from which the claimed inequality follows.

(c) We use the same 'complement rewriting idea' as for (b): noting that $\mathcal{I}_2 \cap \mathcal{D}^c$ is an increasing event, Harris inequality implies that

$$\begin{aligned}\Pr(\mathcal{I}_1 \cap \mathcal{I}_2 \cap \mathcal{D}) &= \Pr(\mathcal{I}_1 \cap \mathcal{I}_2) - \Pr(\mathcal{I}_1 \cap \mathcal{I}_2 \cap \mathcal{D}^c)\\ &\leq \Pr(\mathcal{I}_1 \cap \mathcal{I}_2) - \Pr(\mathcal{I}_1) \cdot \Pr(\mathcal{I}_2 \cap \mathcal{D}^c).\end{aligned}$$

Using independence of $\mathcal{I}_1$ and $\mathcal{I}_2$, it now follows that

$$\begin{aligned}\Pr(\mathcal{I}_1 \cap \mathcal{I}_2 \cap \mathcal{D}) &\leq \Pr(\mathcal{I}_1) \cdot \Pr(\mathcal{I}_2) - \Pr(\mathcal{I}_1) \Pr(\mathcal{I}_2 \cap \mathcal{D}^c)\\ &= \Pr(\mathcal{I}_1) \cdot \Big(\Pr(\mathcal{I}_2) - \Pr(\mathcal{I}_2 \cap \mathcal{D}^c)\Big) = \Pr(\mathcal{I}_1) \cdot \Pr(\mathcal{I}_2 \cap \mathcal{D}),\end{aligned}$$

from which the claimed inequality follows.