# 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 (Making bags of candy).** The owner of the store CandyLand receives supplies every morning, and immediately sets the candy to be sold. This is done by creating identical bags, each of size $S$. The store receives $n$ different types of candy, and a finite amount of each type.

Your input is: integer $S$, representing the size of each bag, and an array of integers `Type` of length $n$. Entry $\texttt{Type}[i]$ is the number of candy of type $i$. Note:

- Each bag must have exactly $S$ pieces of candy, and the same number of each type.
- The objective is to maximize the number of bags the salesman can make.
- The optimal solution does not need to use all the candy, nor all the types of candy.
- Example: for $S = 2$ and $\texttt{Type}= [10, 12, 14]$ the answer is 12, as we can make these many bags with one piece of types 2 and 3 each.
- Example: for $S = 3$ and $\texttt{Type}= [10, 12, 14]$ the answer is 10, putting one piece of each type in every bag.

**(a)** (6 points) Design a polynomial time algorithm to find the maximal number of bags the salesman can make.

**(b)** (4 points) Suppose we also require that each bag must have at least $k$ different types of candy. Design a polynomial time algorithm to find the maximal number of bags the salesman can make.

Explain why your algorithm works and analyze its running time in terms of the input size. You may assume that standard mathematical operations take $O(1)$.

**Solution.**

**(a)** Note that if you can make $T$ bags of candy, then you can also do $T - 1$, by using the same configuration and having more leftover. Thus, the optimal value $T^{opt}$ satisfies the following property: *we can make $T$ identical bags if and only if $T \leq T^{opt}$.* Clearly,

$$T^{opt} < \max\{\texttt{Type}\} + 1$$

as more than that implies we will have empty bags. These observations lead to a binary search algorithm: to check if $T$ bags can be made, for a given $T$, we add the entries of $\left\lfloor \frac{\texttt{Type}[i]}{|T|} \right\rfloor$ and compare this number to $S$. If the sum is bigger than $S$, we move up (smaller values of $T$ lead to larger bags), otherwise we move down. By the first observation, the threshold value found by the binary search is $T^{opt}$.

For the running time, notice that the number of rounds is $\log(\max\{\texttt{Type}\})$, and in each round we do $O(n)$ operations, so the running time is $O(n \log(M))$ (where $M$ is the maximal of the entries of `Types`), which is certainly polynomial in the input size.

**(b)** We can modify the previous idea by setting a counter for the types of candy we will have in our bags. Before adding $\left\lfloor \frac{\texttt{Type}[i]}{|T|} \right\rfloor$, we check if it is positive and if so we increase the counter. The binary branching rule is as before but also includes the condition that the counter must be at least $k$ to move up. As this introduces $O(n)$ operations in each round, our running time is still the same.

## 2. Theory of Linear Inequalities

**Problem**. Given polytopes $P \subseteq \mathbb{R}^n$ and $Q \subseteq \mathbb{R}^{n+q}$ such that $P = proj_x(Q)$ be the projection of $Q$, i.e.

$$P = \{x \in \mathbb{R}^n : \exists y \in \mathbb{R}^q, (x, y) \in Q\}.$$

Then show that

1. (5 points) $Q$ has at least as many faces as $P$.
2. (3 points) The number of facets of $Q$ is at least the logarithm (to base 2) of the number of faces of $P$.
3. (2 points) Consider the permutahedron $\Pi_n \subset \mathbb{R}^n$, the convex hull of the set $S_n$ of the $n!$ vectors that can be obtained by permuting the entries of the vector $(1, 2, \ldots, n-1, n)$. It is easy to check that each permutation of $(1, 2, \ldots, n)$ gives a vertex of $\Pi_n$ (You need not prove this fact). If $U$ is a polytope such that $proj_x(U) = \Pi_n$ show that $U$ must have at least $\Omega(n \log n)$ facets.

**Solution.**

1. Let $F$ be a face of $P$, i.e. $F = \{x \in P : c^T x = \delta\}$ for some $c \in \mathbb{R}^n$ where $\delta = \max\{c^T x : x \in P\}$. Now consider the cost function $\tilde{c} = (c, 0) \in \mathbb{R}^{n+q}$. We have

   $$\begin{aligned} \max\{\tilde{c}^T(x,y) : (x,y) \in Q\} &= \max\{c^T x : (x,y) \in Q\} \\ &= \max\{c^T x : x \in P\} = \delta \end{aligned}$$

   Thus $F' = \{(x, y) \in Q : \tilde{c}^T(x, y) = \delta\}$ is a face of $Q$.

   We now show that $proj_x(F') = F$. Observe that if $(x, y) \in F' \subseteq Q$ then $x \in P$. Moreover, $c^T x = \tilde{c}^T(x, y) = \delta$ and thus $x \in F$ giving $proj_x(F') \subseteq F$. For the other side, let $x \in F \subseteq P$. Thus there exists $y \in \mathbb{R}^q$ such that $(x, y) \in Q$. Moreover, $\tilde{c}^T(x, y) = c^T x = \delta$. Thus $(x, y) \in F'$. This implies that $F \subseteq proj_x(F')$ giving us the equality. Thus we have a map from faces of $P$ to faces of $Q$ that is injective completing the proof.

2. Let $Q$ have $m$ facets. Then $Q$ has at most $2^m$ faces since every face of $Q$ is obtained by intersection of some subset of facets of $Q$. But the number of faces of $P$ is at most the number of faces of $Q$. Thus the number of faces of $P$ is at most $2^m$ or equivalently, the number of facets of $Q$ is at least the logarithm of the number of faces of $P$.

3. Since $\Pi_n$ has $n!$ vertices, it has at least $n!$ faces and therefore any $U$ s.t. $proj_x(U) = \Pi_n$ at least $log(n!) = \Omega(n \log n)$ facets.

## 3. Graph Theory

**Problem**. Let $G$ be a 4-connected graph and $s_1, s_2, s_3, s_4$ be four distinct vertices of $G$, and assume that $G - \{s_1, s_2, s_3, s_4\}$ contains a subdivision of $K_6$. Show that $G$ contains two vertex disjoint paths, one from $s_1$ to $s_3$ and the other from $s_2$ to $s_4$.

**Solution.** Let $T$ denote a subdivision of $K_6$ in $G - \{s_1, s_2, s_3, s_4\}$ and let $t_1, \ldots, t_6$ denote its branch vertices. Moreover, let $Q_{ij}$ denote the branch paths in $T$ between $t_i$ and $t_j$, for all distinct $1 \leq i, j \leq 6$.

Since $G$ is 4-connected, $G$ contains four pairwise disjoint paths from $\{s_1, s_2, s_3, s_4\}$ to $\{t_1, \ldots, t_6\}$, using exactly four vertices from $\{t_1, \ldots, t_6\}$. (This follows from Menger's theorem.) We choose such four paths, say $P_1, P_2, P_3, P_4$, to minimize the number of edges that are in those paths and outside $T$. Without loss of generality, assume $P_i$ is between $s_i$ to $t_i$ for $i = 1, 2, 3, 4$.

We claim that for each $1 \leq i \leq 4$, $Q_{i5} - t_i$ is disjoint from $P_1 \cup P_2 \cup P_3 \cup P_4$. For, otherwise, let $v \in V(Q_{i5} - t_i) \cap V(P_1 \cup P_2 \cup P_3 \cup P_4)$ such that $vQ_{i5}t_5$ is minimal. Note that $v \in V(P_k)$ for some $1 \leq k \leq 4$, and let $P'_k = s_k P_k v \cup v Q_{i5} t_5$ which is a path from $s_k$ to $t_5$ and disjoint from $(P_1 \cup P_2 \cup P_3 \cup P_4) - P_k$ and $\{t_1, t_2, t_3, t_4, t_6\}$. Let $P'_i = P_i$ for $i \neq k$. Clearly, the number of edges in $P'_1, P'_2, P'_3, P'_4$ and outside $T$ is smaller than the the number of edges in $P_1, P_2, P_3, P_4$ and outside $T$, a contradiction.

Similarly, for each $1 \leq i \leq 4$, $Q_{i6} - t_i$ is disjoint from $P_1 \cup P_2 \cup P_3 \cup P_4$.

Now $P_1 \cup Q_{15} \cup Q_{35} \cup P_3$ and $P_2 \cup Q_{26} \cup Q_{46} \cup P_4$ are disjoint paths from $s_1$ to $s_3$ and from $s_2$ to $s_4$, respectively.

# 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 (Advanced) Algorithms

**Problem**. Let $A$ be an $n \times m$ integer matrix. Consider the matrix

$$P = A(A^\top A)^+ A^\top\,,$$

where for any matrix $B$, $B^+$ denotes the pseudo-inverse of $B$. (If $B = UDV^T$ is the SVD of $B$, then $B^+ = VD^{-1}U^T$.) Show that

1. $P^2 = P\,.$
2. The eigenvalues of $P$ are all 0 or 1.
3. $P$ is the orthogonal projection to the span of the columns of $A$ (i.e., $Px = x$ if $x$ is in the span of the columns of $A$ and $Px = 0$ if it is orthogonal to the span.)
4. The leverage score of a row $a_i$ of $A$
$$\sigma_i = a_i(A^\top A)^+ a_i^\top$$
satisfies $0 \le \sigma_i \le 1$.
5. $\sigma_i(A) = \left\|A(A^\top A)^+ A^\top e_i\right\|_2^2$.
6. Let
$$\widetilde{\sigma}_i = \left\|RA(A^\top A)^+ A^\top e_i\right\|_2^2$$
where $R$ is a random $k \times n$ matrix with iid Gaussian entries from $N(0, \frac{1}{k})$. How well does $\widetilde{\sigma}_i$ approximate $\sigma_i$ for every $i$? What should $k$ be to guarantee a $1+\epsilon$ relative error approximation whith high probability?
7. Given an oracle for solving linear systems, how would you compute $\widetilde{\sigma}_i$ efficiently?

**Solution.**

1. Direct calculation, using $B^+BB = B$ where $B = A^T A$.
2. If the eigenvalues of $P$ are $\lambda_i$, then the eigenvalues of $P^2$ are $\lambda_i^2$. Since $\lambda_i^2 = \lambda_i$, the conclusion follows.
3. We have $PA = A$, so the columns of $A$ are preserved. Moreover, for any $x$ orthogonal to all columns of $A$, $A^\top x = 0$, so $Px = 0$.
4. We can write
$$\sigma_i = e_i^\top (A(A^\top A)^+ A^\top) e_i = e_i^\top P e_i = e_i^\top P^2 e_i = \|Pe_i\|^2 \le 1$$
since all eigenvalues of $P$ are at most 1.

5. Note that
$$\left\|A(A^\top A)^+ A^\top e_i\right\|_2^2 = \|Pe_i\|^2$$
and use the previous step.

6.
$$\mathbf{E}\widetilde{\sigma}_i = \mathbf{E}(e_i^\top PR^\top RPe_i) = e_i^\top P\mathbf{E}(R^\top R)Pe_i = e_i^\top P^2 e_i = \sigma_i$$
where we used $\mathbf{E}(R^\top R) = I$. Consider the vectors $v_i = RPe_i$. Then $\widetilde{\sigma}_i = \|RPe_i\|^2$. Moreover, by the Johnson-Lindenstrauss Lemma, for any $v$,
$$\Pr\left(\left|\|Rv\|^2 - \|v\|^2\right| > \epsilon\|v\|^2\right) < 2e^{-\epsilon^2 k/4}$$
Therefore using $k = O(\log n/\epsilon^2)$, and applying the above to every $v_i$, we get that with high probability for all $i$,
$$(1-\epsilon)\widetilde{\sigma}_i \le \|v_i\|^2 \le (1+\epsilon)\widetilde{\sigma}_i.$$

7. To compute $RA(A^\top A)^+$, we can view it as the solution of $k$ linear systems $r_iA = (A^\top A)x$. The run time would then be the time to solve $k$ such systems plus the time to multiply a $k \times n$ matrix with an $n \times m$ matrix (to compute $RA$) and to multiply a $k \times m$ matrix with an $m \times n$ matrix (to finish).

## 5. Combinatorial Optimization

**Problem**.

(i) (4 points) Let $E$ be a finite set and $f : 2^E \to \mathbb{R}_+$ a submodular function with $f(\{e\}) \le 2$ for all $e \in E$. We have an oracle $\mathcal{O}_1$ to find the maximum cardinality set $X \subseteq E$ with $f(X) = 2|X|$, for any such given submodular function $f$ (given by a value oracle). This problem is called the Polymatroid Matching Problem.

Disjoint Pairs Problem: Let $E_1, \ldots, E_k$ be pairwise disjoint unordered pairs and let $(E, \mathcal{F})$ be a matroid (given by an independence oracle), where $E = E_1 \cup E_2 \cup \ldots \cup E_k$. The Disjoint Pairs Problem is to find the maximum cardinality set $I \subseteq \{1, \ldots, k\}$ such that $\bigcup_{i \in I} E_i \in \mathcal{F}$.

Give a polynomial time algorithm (in the size of the ground set $E$) using oracle $\mathcal{O}_1$ (i.e., each call to the oracle is counted as a single step) to solve the Disjoint Pairs Problem. In other words, reduce the Disjoint Pairs Problem to the Polymatroid Matching Problem polynomially.

(ii) (6 points) Now assume that you have an oracle $\mathcal{O}_2$ to solve the DISJOINT PAIRS PROBLEM.

Consider a graph $G = (V, E)$ with each edge belonging to one color class: red, blue and green. We want to find the maximum cardinality acyclic subset of edges so that the number of edges of any single color does not exceed $\lceil n/3 \rceil$. Show that this is a matroid intersection problem (2 points).

Reduce matroid intersection to the DISJOINT PAIRS PROBLEM polynomially, i.e., give a polynomial time algorithm to find maximum cardinality set in the intersection of two matroids using the given oracle $\mathcal{O}_2$. (4 points). Assume oracle access to the rank functions of the matroids as well.

**Solution.**

(i) Construct $f(T) = r(\bigcup_{i \in T} E_i)$, for $T \subseteq \{1, \dots, k\}$ and $r(\cdot)$ be the rank function of the matroid. Note that $f(T) = 2|T|$ if and only if $\bigcup_{i \in T} E_i \in \mathcal{F}$ since $E_i$ are pairwise disjoint pairs. Moreover, $f(\{i\}) = r(E_i) \leq 2$ for all $i \in \{1, \dots, k\}$. Therefore, a single oracle call to the first problem solves the second problem.

(ii) This is a matroid intersection problem: graphic matroid and partition matroid, wherein each color class has an upper bound of $\lceil n/3 \rceil$ (2 points). We will show that matroid intersection reduces to the DISJOINT PAIRS PROBLEM.

Let the two matroids be $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$. Construct a new matroid $M' = (E \cup E', \mathcal{F})$ with the ground set $E$ duplicated, such that a subset $S = W \cup W'$ with $W \subseteq E$, $W' \subseteq E'$, is independent in the matroid $M'$ if corresponding subset of elements with the first copy, i.e., $W$, are independent in the first matroid $M_1$ and the subset $W'$ with the second copies is independent in the second matroid. Check that this is a valid matroid (subset property holds independently in each copy, and growth property holds as well since either the corresponding subset in the first matroid is smaller or the second matroid is smaller). Let $E_i = (e_i, e_i')$ for each edge $e_i \in E$ in the given graph. Then, the matroid intersection problem is to find a subset $T \subseteq \{1, \dots, n\}$ of maximum cardinality such that $r(\bigcup_{i \in T} E_i) = 2|T|$, i.e., the subset of first copies is equal to subset of second copies and the subset is independent in both the matroids.

## 6. Probabilistic Methods

**Problem**. Suppose that there are $k$ people in a lift at the ground floor, and that each wants to get off at a random floor of one of the $n$ upper floors (to clarify: all people make their choices independently). Let $X$ denote the number of lift stops, i.e., total number of distinct floors chosen by the $k$ people. For $k = k(n)$ satisfying $k \to \infty$ and $k/n \to 0$ as $n \to \infty$, show that the expectation and variance of $X$ satisfy $\mathbf{E}X \sim k$ and $\operatorname{Var} X = o((\mathbf{E}X)^2)$.

**Hints:**

- Working with the 'complement' $Y := n - X$ might be easier.

- Recall that $(1-x)^r = 1 - rx + O((rx)^2)$ and $(1-x)^r = 1 - rx + \binom{r}{2}x^2 + O((rx)^3)$ when $r \geq 2$ and $x \in [0,1]$.

**Solution.** For each $1 \leq i \leq n$, let $I_i$ denote the indicator variable for the event that none of the $k$ people gets off at floor $i$. Note that

$$Y = n - X = \sum_{1 \leq i \leq n} I_i.$$

For the expected value, note that by independence of the choices we have

$$\mathbb{P}(I_i = 1) = \big(1 - 1/n\big)^k.$$

We thus obtain

$$\mathbf{E}X = n - \mathbf{E}Y = n - \sum_{1 \leq i \leq n} \mathbb{P}(I_i = 1) = n \cdot \Big(1 - \big(1 - 1/n\big)^k\Big).$$

Using the hint $(1-x)^r = 1 - rx + O((rx)^2)$ we see that

$$\mathbf{E}X = n \cdot \Big(1 - \big(1 - 1/n\big)^k\Big) = n\Big(k/n + O\big((k/n)^2\big)\Big) = k + O(k^2/n) \sim k.$$

For the variance, in view of $X = n - Y$ and $Y = \sum_{1 \leq i \leq n} I_i$ it follows that

$$\operatorname{Var} X = \operatorname{Var} Y = \sum_{1 \leq i,j \leq n} \Big(\mathbb{P}(I_i = 1 \text{ and } I_j = 1) - \mathbb{P}(I_i = 1)\mathbb{P}(I_j = 1)\Big).$$

When $i = j$, then we have

$$\mathbb{P}(I_i = 1 \text{ and } I_j = 1) = \mathbb{P}(I_i = 1).$$

When $i \neq j$, then by independence of the choices we have

$$\mathbb{P}(I_i = 1 \text{ and } I_j = 1) = \big(1 - 2/n\big)^k.$$

Putting these case distinctions together, we see that

$$\operatorname{Var} X \leq \mathbf{E}X + n(n-1) \cdot \Big(\big(1 - 2/n\big)^k - \big(1 - 1/n\big)^{2k}\Big).$$

Using the hint $(1-x)^r = 1 - rx + \binom{r}{2}x^2 + O((rx)^3)$, the point is that

$$\begin{aligned}\big(1 - 2/n\big)^k - \big(1 - 1/n\big)^{2k} &= \binom{k}{2}\left(\frac{2}{n}\right)^2 - \binom{2k}{2}\left(\frac{1}{n}\right)^2 + O\big((k/n)^3\big) \\ &\leq O\big(k/n^2\big) + O\big((k/n)^3\big) = o(k^2/n^2).\end{aligned}$$

Recalling that $\mathbf{E}X \sim k \to \infty$, it thus follows that

$$\operatorname{Var} X \leq \mathbf{E}X + o(k^2) = o\big((\mathbf{E}X)^2\big).$$