General Instructions  The following assignment is meant to be challenging. Feel free to discuss with fellow students, though please write up your solutions independently and acknowledge everyone you discussed the homework with on your writeup. I also expect that you will not attempt to consult outside sources, on the Internet or otherwise, for solutions to any of these homework problems. Finally, please provide a formal mathematical proof for all your claims.

Problem 1. (14 points)
In this problem, we will show how implement Caratheodory’s theorem constructively, assuming only that the relevant extreme points form a solvable polytope. Specifically, given a polytope $P \subseteq \mathbb{R}^n$ represented by a polynomial-time separation oracle, and a point $x \in \mathbb{R}^n$, we will show how to efficiently express $x$ as a convex combination of at most $n + 1$ vertices of $P$, or else declare that $x \not\in P$. Note that $P$ may have an exponential (in $n$) number of vertices and facets.

(a) [4 points]. Let $V \subseteq \mathbb{R}^n$ denote the set of all vertices of $P$. Consider the following linear program, with variables $\{\lambda_v : v \in V\}$.

$$
\begin{align*}
\text{minimize} & \quad 0 \\
\text{subject to} & \quad \sum_{v \in V} \lambda_v \cdot v = x \\
& \quad \sum_{v \in V} \lambda_v = 1 \\
& \quad \lambda \succeq 0
\end{align*}
$$

(1)

Show that $x \in P$ if and only if the above linear program is feasible. Moreover, show that extreme-point solutions of the linear program correspond to expressions of $x$ as a convex combination of at most $n + 1$ vertices of $P$.

(b) [10 points]. Unfortunately, linear program (1) has too many variables — one per vertex of $P$. Show that, regardless of the number of vertices of $P$, you can nevertheless solve (1) in $\text{poly}(n, b)$ time, where $b$ is an upperbound on the bit-complexity of each vertex of $P$. Your algorithm should either output a list of at most $n + 1$ vertices $V' \subseteq V$ along with coefficients $\{\lambda_v : v \in V'\}$ such that $\sum_{v \in V'} \lambda_v \cdot v = x$, or else correctly declare that $x \not\in P$. Recall that the only assumption we make about $P$ is that we have access to an implementation of a separation oracle for $P$, which you may assume runs in time polynomial in $n, b$, and the bit complexity of the input test point. (Hint: Use LP duality, equivalence of separation and optimization, and complementary slackness)

Problem 2. (12 points)
In this problem, we will examine some miscellaneous properties of matroids. We use $\mathcal{M} = (\mathcal{X}, \mathcal{I})$ to denote an arbitrary matroid.
(a) [4 points]. Prove the strong exchange property of matroids: If $B$ and $B'$ are bases of matroid $M$, then for every $i \in B \setminus B'$ there is some $j \in B' \setminus B$ such that $B \setminus \{i\} \cup \{j\}$ is a basis of $M$. Show that the exchange property is implied by the strong exchange property for every downwards-closed set system. (i.e. we could have equivalently defined matroids using the strong exchange property instead).

(b) [2 points]. Show that matroids are closed under truncation. Namely, if $M = (\mathcal{X}, \mathcal{I})$ is a matroid, and $k$ is a nonnegative integer, then the set system $M_k = (\mathcal{X}, \mathcal{I}_k)$ with $\mathcal{I}_k = \{S \in \mathcal{I} : |S| \leq k\}$ is also a matroid.

(c) [3 points]. Recall that we call a set $C \subseteq \mathcal{X}$ a circuit of the matroid $M$ if it is a minimal dependent set in $M$. Note that a circuits of a graphical matroid correspond to cycles in the graph. Prove the circuit property of matroids: if $M = (\mathcal{X}, \mathcal{I})$ is a matroid with distinct weights $w \in \mathbb{R}^\mathcal{X}$ assigned to its ground set, and $C$ is a circuit of $M$, then the element $i \in C$ of minimum weight is not in any maximum-weight basis of $M$.

(d) [3 points]. We call a set $C \subseteq \mathcal{X}$ a cut of the matroid $M$ if every basis of $M$ intersects $C$, and no proper subset of $C$ has this property. Note that a cut of a graphical matroid correspond to our notion of a cut in a graph — namely, a family of edges who’s removal disconnects some connected component of the graph. Prove the cut property of matroids: if $M = (\mathcal{X}, \mathcal{I})$ is a matroid with distinct weights $w \in \mathbb{R}^\mathcal{X}$ assigned to its ground set, and $C$ is a cut of $M$, then the element $i \in C$ of maximum weight is in every maximum-weight basis of $M$.

Problem 3. (16 points)
Given two matroids $M_1 = (\mathcal{X}, \mathcal{I}_1)$ and $M_2 = (\mathcal{X}, \mathcal{I}_2)$ with a common ground set $\mathcal{X}$, and a weight vector $w \in \mathbb{R}^\mathcal{X}$, the maximum-weight common independent set of $M_1$ and $M_2$ is the set $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ maximizing $\sum_{i \in S} w(i)$. We discussed in class how the maximum-weight common independent set can be computed in time poly($|\mathcal{X}|$), given only independence-oracle access to the matroids $M_1$ and $M_2$. Recall that this followed from the integrality of the polytope $P(M_1) \cap P(M_2)$, where $P(M_1)$ and $P(M_2)$ are the two matroid polytopes.

We will now consider a related problem, the minimum-weight common-basis problem, show how to solve efficiently, and show that it can be viewed as a generalization of the shortest path problem in directed graphs. We will then use this insight to show that the shortest path problem subject to a cardinality constraint is also solvable in polynomial time. Given two matroids $M_1 = (\mathcal{X}, \mathcal{I}_1)$ and $M_2 = (\mathcal{X}, \mathcal{I}_2)$, let $\mathcal{B}_1 \subseteq 2^\mathcal{X}$ denote the family of all bases (maximum-cardinality independent sets) of $M_1$, and similarly $\mathcal{B}_2$ for $M_2$. Given a weight vector $w \in \mathbb{R}^\mathcal{X}$, the minimum-weight common basis of $M_1$ and $M_2$, if it exists, is the set $B \in \mathcal{B}_1 \cap \mathcal{B}_2$ minimizing $\sum_{i \in B} w(i)$. Note that for a common basis to exist, both matroids must have the same rank (though this is not sufficient).

(a) [6 points]. Show that the minimum-weight common-basis problem can be solved in polynomial time in $|\mathcal{X}|$, given only independence-oracle access to the two matroids $M_1$ and $M_2$. Your algorithm should either output a minimum-weight common-basis of $M_1$ and $M_2$ if one exists, or else assert that there is no common basis.

(Hint: You can start by showing that that the convex hull of common bases is a face of the matroid intersection polytope)
(b) [6 points]. Show that the shortest path problem in weighted directed graphs can be reduced, in linear time, to the minimum-weight common-basis problem when the graph has no negative cycles.

(Hint: Add self-loops to all nodes other than the source and destination, and define two partition matroids such that their common bases are precisely a shortest path combined with some of the self-loops.)

(c) [4 points]. Given an integer \( k \) and a weighted directed graph, show that finding the shortest path with at most \( k \) edges can also be reduced in linear time to the minimum-weight common-basis problem when the graph has no negative cycles.

(Hint: Use Problem 2b.)

Problem 4. (8 points)
The following is a (simplification of) a task encountered in speech recognition. Here, we wish to select a small vocabulary (a set of words) which express many common phrases. Specifically, you are given a set \( P \) of phrases which, collectively, use only the words in the set \( W \). Moreover, each phrase \( p \in P \) is given a weight \( w(p) \) indicating its importance. For a vocabulary \( V \subseteq W \), we use \( P(V) \subseteq P \) to denote the phrases which use only words in \( V \). Your goal is to choose a vocabulary optimizing a tradeoff between the total weight of expressed phrases and the size of the vocabulary; specifically, find \( V \subseteq W \) maximizing \( \left( \sum_{p \in P(V)} w(p) \right) - |V| \). Show that this problem is solvable in poly(|\( P \)|, |\( W \)|) time. You may assume that you can test whether a particular word occurs in a particular phrase in constant time.

Problem 5. (12 points)
Sensor networks are frequently deployed for sensing the environment (e.g. temperature or pressure). Consider the following simplified scenario: There are \( n \) sensors \( S \) and \( m \) reference locations \( L \). For \( \ell \in L \) and \( s \in S \), you are given a coverage quality \( q(s, \ell) \in \mathbb{R}_+ \). We assume there are \( T \) time periods, and your goal is to select a subset of the sensors \( X_t \subseteq S \) to “activate” during each time period \( t \in \{1, \ldots, T\} \). The quality of the resulting schedule \( (X_1, \ldots, X_T) \) is given by \( \sum_{t \in L} \max \{ q(s, \ell) : s \in \bigcup_{i=1}^{T} X_i \} \).

You are asked to compute the maximum-quality schedule, subject to two constraints: (1) due to communication/bandwidth limitations, no more than a given number \( k \) of the sensors may be active in any time period; (2) Due to power limitations, each sensor \( s \) can only operate in time periods \( F_s \subseteq \{1, \ldots, T\} \), where \( F_s \) is given. Your input consists of \( n, m, T, k \), the coverage qualities \( q(s, \ell) \) for each \( s \in [n] \) and \( \ell \in [m] \), and the set \( F_s \subseteq [T] \) for each \( s \in [n] \).

(a) [4 points]. Show that this problem is NP-hard.

(b) [8 points]. Come up with an algorithm which runs time polynomial in \( n, m \) and \( T \), and finds a schedule approximately maximizing quality. Aim for the best (multiplicative) approximation ratio that you can get.