Problem 1: This problem is about speeding up Ford-Fulkerson by picking the augmenting path in each step not arbitrarily, but by choosing the one with the largest minimum capacity amongst all its edges.

(a) (2 points) Prove the following: in a directed and capacitated graph any flow $f$ can be decomposed into the sum of at most $m$ flows $f_1, \ldots, f_r$, $r \leq m$, such that each $f_i$ is supported on either a path or a cycle.

(b) (3 points) Given a directed graph with arbitrary edge weights (could be positive or negative), and a start vertex $s$ and end vertex $t$, we would like to find the path from $s$ to $t$ whose minimum weight is as large as possible. Show that this problem can be solved in $O(m + n \log n)$. Hint: modify Dijkstra's algorithm.

(c) (5 points) In class we said the Ford Fulkerson $s$-$t$ max flow algorithm repeatedly finds augmenting paths in residual graphs. For an augmenting path $P$ define $c(P)$ as the minimum capacity on that path (in the residual graph). Suppose we augment along an augmenting path $P$ which has maximum $c(P)$ value. Use (a) and (b) to show that the resulting max flow algorithm would run in time $O((m + n \log n)m \log(f^*)) = O((m + n \log n)m \log(mU))$, where $U$ is the largest capacity.

Problem 2: In class we didn’t discuss the bit complexity in detail for interior point methods. Here you will fill in some of the gaps.

(a) (2 points) For $A, b, c$ having integer entries with $A \in \mathbb{R}^{m \times n}$, consider the linear program

$$\text{min } c^T x$$

$$s.t. \quad Ax \geq b$$

Define $L = C(1 + \log(1 + d_{max}) + \log(1 + \max\{\|b\|_\infty, \|c\|_\infty\}))$ for sufficiently large constant $C$. Here $d_{max}$ is the largest absolute value of a determinant of a submatrix of $A$. Show that if (I) is bounded then there is an optimum solution $x^*$ to this LP s.t. for all $i$ $x_i^*$ requires $\leq L$ bits of precision. That is, for all $i$ the value $x_i^*$ is rational and can be expressed as $\alpha_i/\beta_i$ for integers $-m2^L \leq \alpha_i, \beta_i \leq m2^L$. 
(b) (5 points) Consider the linear program (I) from (a). Consider the transformation to a new LP as follows:

\[
\begin{align*}
\min & \quad c^T x + nC' \cdot 2^{CL} z \\
\text{s.t.} & \quad Ax + \mathbb{1} z \geq b \\
& \quad 0 \leq z \leq 2^L \\
& \quad \forall i, -2^{L+1} \leq x_i \leq 2^{L+1}
\end{align*}
\]

(II)

where \( \mathbb{1} \) is the all-ones vector. We noted that \( x = 0, z = \|b\|_\infty \) is an obvious feasible solution of (II). Show that for some constants \( C, C' > 0 \) (I) is bounded and feasible with optimal solution \( x^* \) iff \((x,z) = (x^*,0)\) is an optimal solution to (II) for some \( x^* \) with \( \|x^*\|_\infty \leq 2^L \).

(c) (4 points) In class we assumed the matrix \( \nabla^2 f_{\lambda k}(x) \) considered in Newton steps was always invertible. Show that this can be ensured throughout the course of the path-following IPM algorithm presented in class to solve (I), at the expense of increasing both the number of rows and columns of \( A \) by an additive \( O(m+n) \) in pre-processing.

(d) (4 points) In the Lecture 17 notes, equations (1)-(3), we show if \( x \) is perfectly central for \( \lambda = \Omega(m/\varepsilon) \), then \( c^T x \leq \text{OPT} + \varepsilon \). However in Lecture 18 we did not show how to find a perfectly central point for large \( \lambda \); rather we showed how to find a finely central point. Show that \( x \) being finely central for \( \lambda = \Omega(m/\varepsilon) \) is sufficient to achieve \( c^T x \leq \text{OPT} + \varepsilon \).

\textbf{Hint:} It may help to first prove a generalized Cauchy-Schwarz inequality, namely that \( \langle x,y \rangle \leq \|x\|_B \cdot \|y\|_{B^{-1}} \) for any real, symmetric positive definite matrix \( B \).

\textbf{Problem 3:} Recall the learning from experts setup. On each day it will either rain or not. \( n \) experts each tell us their opinion, and we must take their opinions as input to form our own opinion and make a prediction. We then find out at the end of the day whether we were right or wrong. If \( E_i \) is the number of mistakes of expert \( i \), and \( M \) is our number of mistakes, then forming our opinions using the Multiplicative Weights algorithm guarantees

\[ M \leq \min_{1 \leq i \leq n} E_i \cdot \eta T + \frac{\ln n}{\eta R} \]

for any \( \eta \leq 1/2 \). In particular if we choose \( \eta = \sqrt{\ln n}/T \) when \( T \geq 2 \ln n \) (so that \( \eta \leq 1/2 \)), then our regret \( R \) is at most \( \sqrt{T \ln n} \).

(a) (5 points) Show that no algorithm (not just Multiplicative Weights) can achieve \( R = o(\sqrt{T}) \) in general for this problem. \textbf{Hint:} Consider the case \( n = 2 \).

(b) (5 points) \textbf{Bonus:} Show that no algorithm can achieve \( R = o(\sqrt{T \ln n}) \) when \( T \) is sufficiently large.
Problem 4: (10 points) Consider the regret setting from Lecture 19, Section 3.1, i.e. there are \( n \) experts, and on each of the \( T \) days \( 1 \leq t \leq T \) there is a cost vector \( m^{(t)} \in [-1, 1]^n \). We must, at the beginning of each day, output a probability vector \( p^{(t)} \) over experts such that

\[
\sum_{t=1}^{T} m^{(t)} \cdot p^{(t)} \leq \min_{1 \leq i \leq n} \sum_{t=1}^{T} m^{(t)} + R
\]

where the regret \( R \) is small. Multiplicative weights (MW) gave the guarantee \( R = O(\sqrt{T \log n}) \), but it made the assumption that at the end of each day \( t \) we learn the entire cost vector \( m^{(t)} \) (which MW needed to update its probability vector). In the real world though, we often can only learn the costs/benefits of actions we perform and we don’t get to learn about what would have happened “if we had only done some other action \( X \) instead” (think about movie “Groundhog Day”; a movie I recommend if you haven’t already seen it). What if we must design an algorithm which on each day comes up with a probability vector \( p^{(t)} \), then chooses an expert randomly according to \( p^{(t)} \), then only learns \( m^{(t)}_i \) for the expert \( i \) that was randomly chosen? It turns out we can still obtain good regret in this model.

Show that if \( T \geq n \) then it is possible to obtain regret \( O(T^{2/3} n^{1/3} (\log n)^{1/3}) \) in the above new model. **Hint:** Break up the \( T \) days into \( k \) blocks of \( T/k \) days each. Start block \( \tau \), where \( 1 \leq \tau \leq k \), with probability vector \( p^{(\tau)} \). Feed some loss vector into MW at the end of block \( \tau \) with \( p^{(\tau)} \) to obtain \( p^{(\tau + 1)} \). Within each block, do a mix of sampling according to \( p^{(\tau)} \) (“exploiting” the knowledge gained so far) as well as sampling all experts more equitably (“exploring” all the options).

Problem 5: (1 point) How much time did you spend on this problem set? If you can remember the breakdown, please report this per problem. (sum of time spent solving problem and typing up your solution)