CSCI 670:
Advanced Analysis of Algorithms Fall 2021

Quiz 3 (Duration: 100 mins)

Name:

Student ID:

Email Address:

Problem 1 (25 points)**.** Let $G = (V, E)$ be directed acyclic graph (DAG), i.e. G is a directed graph where no directed paths consist of a cycle. A path P is denoted by a sequence of vertices $v_1, v_2, \ldots v_k$ where $(v_i, v_{i+1}) \in E$. We say that a collection $\mathcal{P}$ of paths partition the graph G if

$$V(P_1) \cap V(P_2) = \emptyset \quad \forall P_1 \neq P_2 \in \mathcal{P} \qquad \text{and} \qquad \cup_{P \in \mathcal{P}} V(P) = V$$

where $V(P)$ denotes the set of vertices appearing in P. Design an algorithm to find a minimum cardinality path partition of the graph G. The cardinality of a partition $\mathcal{P}$ is the number of paths in the partition.

Solution 1. First of all, this problem should remind you the box stacking problem we have in the homework 4.

Observe that we can decompose the graph into length 0 paths where each of them consists of a single vertex. With this way we can easily partition the graph into paths with $|V|$ many paths. Since we want to minimize the number of paths in the partition, we can maximize the total number of edges used in the path partition. Since there is at most one edge going out of a vertex and at most one edge going into a vertex, we can formulate this problem as a bipartite matching problem.

Let n be the number of vertices in G. We build a graph $G' = (X, Y, E')$ where $X = \{x_1, \ldots x_n\}$, $Y = \{y_1, \ldots y_n\}$, and $E' = \{(x_i, y_j) \mid (i, j) \in E\}$, i.e. there is an edge from i to j in E.

Now, we will observe the one-to-one correspondance with matchings of G' and path partitions of G. Let M be a matching of G', then consider the edges $L = \{(i, j) \mid (x_i, y_j) \in M\}$. Since each vertex of Y is matched to at most one vertex from X, there can be at most one edge going inwards to a vertex in L. One can similarly show that there can be at most one edge going outwards from a vertex. Thus, edges in L consists of a collection paths. Similary, any path partition $\mathcal{P}$ can be converted into matching of M.

Let e be the number of edges in a path partition $\mathcal{P}$ (or equivalently a matching M), then $n - e = |\mathcal{P}|$ is the number of paths in the partition. Therefore, maximizing number of edges used in $\mathcal{P}$ (or M) also minimizes the cardinality of partition. Maximum mathcing in a bipartite graph can be find by Ford-Fulkerson algorithm once we represent the problem as s-t flow.

Problem 2 (25 points)**.** Consider a two-player zero-sum game given by a matrix $A \in \mathbb{R}^{m \times n}$. The first player has m pure strategies $\{1, \ldots, m\}$, while the second player has pure n strategies $\{1, \ldots, n\}$. Each player can randomize her strategy (mixed strategy). A randomized strategy is just a probability distribution over pure strategies and is called mixed strategy. Let the probability distribution $p = (p_1, \ldots, p_m)$ denote the mixed strategy of the first player, and $q = (q_1, \ldots, q_n)$ denote that of the second player. In the only round of the game the first player selects a mixed strategy p, after which the second player selects a mixed strategy q observing p. The payoff is defined as $p^T Aq$. The objective of the first player is to minimize the payoff, while the objective of the second player is to maximize it. We can formalize this game as minimax program:

$$
\begin{aligned}
\min_p \max_q \quad & p^T A q & \\
\text{s.t.} \quad & p_i \geq 0, & i = 1, \ldots, m \\
& q_j \geq 0 & j = 1, \ldots, n \\
& \sum_{i=1}^{m} p_i = 1 \qquad \sum_{j=1}^{n} q_j = 1 &
\end{aligned}
$$

Show that finding the best strategy of the first player can be found in polynomial-time using linear programming.

Solution 2. We start with an observation. Once the mixed strategy p of the first player is determined, $\max_q p^T Aq = \max_{i \in \{1,\ldots,n\}} p^T A_i$ where A_i is the i^{th} column of A. Therefore, we can say that for any mixed strategy p of the first player, the second player has a pure optimal stategy once she knows p. We will apply this observation to rewrite the mathematical programming.

$$
\begin{aligned}
\min_p \quad & \max\{p^T A_1, p^T A_2, p^T A_2, \ldots p^T A_n\} & \\
\text{s.t.} \quad & p_i \geq 0, & i = 1, \ldots, m \\
& \sum_{i=1}^{m} p_i = 1 &
\end{aligned}
$$

As we proved in the Homework 5, we can rewrite this mathematical program as a linear programm as follows.

$$
\begin{aligned}
\min_{p,\xi} \quad & \xi & \\
\text{s.t.} \quad & p_i \geq 0, & i = 1, \ldots, m \\
& \xi \geq p^T A_i, & i = 1, \ldots, n \\
& \sum_{i=1}^{m} p_i = 1 &
\end{aligned}
$$

Problem 3 (25 points)**.** Let $C_n = (V, E)$ be a cycle graph such that $V = \{0, 1, \ldots n-1\}$ and $(i, i+1 \mod n) \in E$. Consider the uniform random walk X_t on C_n which starts from the vertex 0. Formally, $\Pr[X_{t+1} = X_t + 1 \mod n] = \frac{1}{2}$ and $\Pr[X_{t+1} = X_t - 1 \mod n] = \frac{1}{2}$. Consider for the random walk X_t for $t \geq 0$, let $H_{i,j}$ be the hitting probability of X_t to j before hitting i. In other words, $H_{i,j}$ is the probability of X_t visits j before visiting i. Compute $H_{i,j}$ for any $i < j$. You can without loss of generality assume that $i, j > 0$ and $n \geq 3$.

Solution 3. Consider for some $i < j$. Since $i < j$, we can represent this walk in an infinite line as follows. The walk starts from 0 and it goes up or down with probability 1/2 independently. We want to compute the probability of hitting $-j$ before hitting i. Equivalently, we can represent this as Gambler's ruin problem we discussed in the homework 5. Where gambler starts with i dolars and wins (or loses) a dolar with probability 1/2 independently. We say that gambler wins the game if she earns $n - j$ additional dolars before losing all her money. As we proved in the homework this probability equals to $\frac{i}{i+n-j}$. Therefore, $H_{i,j} = \frac{i}{i+n-j}$.

Problem 4 (25 points)**.** In this problem we will work on a variant of the minimum set-cover problem. Let $G = \{1, 2, \ldots n\}$ be a ground set and $S_1, \ldots S_m$ be subsets of the ground set G. We say that a set C covers S_k if $C \cap S_k \neq \emptyset$. Given sets $S_1, S_2, \ldots, S_m \subseteq G$, find the smallest subset $C \subseteq G$ such that C covers S_k for any $k \in \{1, \ldots, m\}$. Design an algorithm that outputs $O(\log n)$ approximate solution with constant probability (say 1/2). (Hint: You may use LP-relaxation + rounding.)

Solution 4. First of all one can show that this problem is equivalent to the original set-cover problem we discussed in the homework 5. Let's analyze this version separetely.

$$\begin{aligned}
\min \quad & \sum_{i \in G} x_i \\
\text{s.t.} \quad & \sum_{i \in S_k} x_i \geq 1 && (1 \leq k \leq m) \\
& 0 \leq x_i \leq 1 && (i \in G)
\end{aligned}$$

Do randomized rounding.

$$\widetilde{x_i} = \begin{cases} 1 & \text{w.p. } x_i \\ 0 & \text{w.p. } 1 - x_i \end{cases}$$

Now, let's compute the probability of a set S_k being covered.

$$\Pr[S_k \text{ is covered}] = 1 - \prod_{i \in S_k} (1 - x_i) \geq 1 - \left(1 - \frac{1}{|S_k|}\right)^{|S_k|} \geq (1 - 1/e)$$

the first inequality follows from the AM-GM inequality. Now, the idea is to applying the same rounding mechanism a couple of times and taking their union.

Assume that we applied the same rounding algorithm T times. Then,

$$\Pr[\text{At least one } S_i \text{ is uncovered}] \leq \sum_{k=1}^{m} \Pr[S_k \text{ is uncovered}] \leq m \cdot \left(\frac{1}{e}\right)^T$$

setting $T = 2 \ln m$ implies that $\Pr[\text{At least one } S_i \text{ is uncovered}] \leq \frac{1}{m}$. Let X be the output of the rounding, notice that $\mathbb{E}[|X|] = T \cdot LP_{OPT} \leq T \cdot OPT = O(\log n) \cdot OPT$. Thus, the rounding algorithm outputs a $O(\log m)$ approximation to set-cover problem with probability $1 - 1/m \geq 1/2$ assuming $m \geq 2$.