Midterm Exam, CS-450: Algorithms II, 2025-2026

Do not turn the page before the start of the exam. This document is double-sided and has 8 pages.

- You are only allowed to have one A4 page written on one side.
- Communication, calculators, cell phones, computers, etc... are not allowed.
- For the open-ended questions, your explanations should be clear enough and in sufficient detail that a fellow student can understand them. In particular, do not only give pseudocode without explanations. A good guideline is that a description of an algorithm should be such that a fellow student can easily implement the algorithm following the description.
- You are allowed to refer to material covered in the lectures, exercise sets and problem sets including algorithms and theorems (without reproving them).

Good luck!

1 Dual Matroid *(29 points)* Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid. Recall that a base of $\mathcal{M}$ is an independent set $B \in \mathcal{I}$ of maximum size. We say that a set $U \subseteq E$ is a *spanning set for* $\mathcal{M}$ if it contains a base of $\mathcal{M}$, i.e., there exists a base B of $\mathcal{M}$ such that $B \subseteq U$. Then, we define the *dual of* $\mathcal{M}$ as $\text{dual}(\mathcal{M}) = (E, \mathcal{I}')$, where

$$\mathcal{I}' = \{S \subseteq E : E \setminus S \text{ is a spanning set for } \mathcal{M}\}.$$

One can show that $\text{dual}(\mathcal{M}) = (E, \mathcal{I}')$ as defined above is a matroid.

Questions **1a** and **1b** are multiple-choice questions. For each of them, select the correct option. Note that each question has **exactly one** correct answer. Wrong answers are **not penalized** with negative points.

1a *(7 pts)* Let $E_1, \dots, E_\ell$ be a partition of a set E and let $\mathcal{M} = (E, \mathcal{I})$ be the partition matroid given by

$$\mathcal{I} = \{X \subseteq E \colon |X \cap E_i| \le k_i \text{ for } i = 1, \dots, \ell\},$$

where $k_i \le |E_i|$ for all i. Let $\text{dual}(\mathcal{M}) = (E, \mathcal{I}')$. Which of the following describes $\mathcal{I}'$?

A. $\mathcal{I}' = \{X \subseteq E : |X \cap E_i| \le |E_i| - k_i \text{ for } i = 1, \dots, \ell\}$
B. $\mathcal{I}' = \{X \subseteq E : |X \cap E_i| \le k_i \text{ for } i = 1, \dots, \ell\}$
C. $\mathcal{I}' = \left\{X \subseteq E : |X| \le \sum_{i=1}^{\ell} k_i\right\}$
D. $\mathcal{I}' = \{X \subseteq E : |X \cap E_i| > k_i \text{ for } i = 1, \dots, \ell\}$

Solution. A. The set of bases of $\mathcal{M}$ is $\{Y \subseteq E : |Y \cap E_i| = k_i \ \forall i\}$. Therefore, for a set $X \subseteq E$, we have that $E \setminus X$ contains a base of $\mathcal{M}$ if and only if $|(E \setminus X) \cap E_i| \ge k_i$ for all i, which is equivalent to the condition that $|X \cap E_i| \le |E_i| - k_i \ \forall i$. □

1b *(7 pts)* Let $G = (V, E)$ be an undirected connected graph and let $\mathcal{M} = (E, \mathcal{I})$ be the graphic matroid, i.e.

$$\mathcal{I} = \{F \subseteq E : F \text{ is acyclic}\}.$$

Let $\text{dual}(\mathcal{M}) = (E, \mathcal{I}')$. Which of the following describes $\mathcal{I}'$?

A. $\mathcal{I}' = \{F \subseteq E : E \setminus F \text{ is acyclic}\}$
B. $\mathcal{I}' = \{F \subseteq E : E \setminus F \text{ is connected}\}$
C. $\mathcal{I}' = \{F \subseteq E : F \text{ contains a cycle}\}$
D. $\mathcal{I}' = \{F \subseteq E : F \text{ is bipartite}\}$

Solution. B. The bases of $\mathcal{M}$ are spanning trees of G, so $E \setminus F$ contains a base of $\mathcal{M}$ if and only if $E \setminus F$ contains a spanning tree, which is equivalent to $E \setminus F$ being connected. $\square$

1c *(15 pts)* Suppose there is a polynomial-time algorithm that implements an independence oracle for $\mathcal{M}$, i.e. given $S \subseteq E$ tells whether $S \in \mathcal{I}$ or not. Give a polynomial-time algorithm that implements an independence oracle for dual($\mathcal{M}$), i.e. given $S \subseteq E$ tells whether $S \in \mathcal{I}'$ or not, and prove its correctness.

Solution. The algorithm first uses the given oracle to implement the greedy algorithm and compute the rank of $\mathcal{M}$. Then, given $S \subseteq E$, it uses the given oracle to implement the greedy algorithm and compute the rank of the submatroid induced by $E \setminus S$. If the two ranks match it outputs yes, and no otherwise. $\square$

2 Maximum Revenue Routing *(20 points)* In a hilly city, the roads were built as one-way streets because of the steep and treacherous terrain. Any road stretches between two junctions, and there are n junctions in total. Alice owns coffee shops located at some of the junctions. Her partner Bob owns warehouses stocked with croissants, located at some of the junctions. Each coffee shop has an **expected daily revenue**, a positive number indicating how much money that particular coffee shop is expected to make.

Each day, Alice can keep a coffee shop open only if a box of fresh croissants is delivered to it from one of Bob's warehouses. To prevent traffic jams, the delivery routes for distinct boxes must be completely **non-overlapping**: on any day, no two boxes can pass through the same junction, including their starting point (a warehouse) and ending point (a coffee shop).

Your task is to help Alice identify which coffee shops can be kept open today (subject to the constraint above), so as to maximize the total expected daily revenue. For any subset T of the coffee shops, Alice can ask Bob: **"is it possible to keep all the shops in T open?"**, and then Bob correctly responds with "yes" or "no" in time $O(\text{poly}(n))$. Give an algorithm that runs in time $O(\text{poly}(n))$ and justify its correctness.

Hint: to solve this problem you may assume any result proved in lectures, exercise sets and problem set I. If you do this, your solution will be quite short.

Solution. We begin by translating the problem into graph language.

Let $G = (V, E)$ be a directed graph, and let $C, W \subseteq V$ be the vertices corresponding to the coffee shops and warehouses respectively. Let $w : C \to \mathbb{R}_+$ denote the expected daily revenue of coffee shops. We say that a set $X \subseteq C$ is *linked to* W if the vertices in X can be reached from W via vertex-disjoint paths. Our goal is to find $C' \subseteq C$ of the maximal total expected daily revenue which is linked to W.

Lemma 2.1 *Define $\mathcal{I}$ to be the collection of subsets of C where $I \in \mathcal{I}$ if I is linked to W. $\mathcal{I}$ is a matroid, called a **gammoid**.*

Proof. Lemma 2.1 was given as an exercise in Homework 1. $\square$

Now that we showed that the collection of subsets of C linked to W, $\mathcal{I}$, is a matroid, we can restate the problem as finding the maximum weight base of $\mathcal{I}$ which, as proven in class, may be done with the greedy algorithm.

Denote the elements of C as $c_1, \ldots, c_{|C|}$.

Algorithm 1 Greedy Algorithm

```
Sort and relabel the elements of C so that w(c_1) ≥ w(c_2) ≥ ... ≥ w(c_|C|)
Initialize the maximum weight base, R, to ∅
for i = 1, ..., |C| do
    if R ∪ {c_i} ∈ I then
        R ← R ∪ {c_i}
    end if
end for
```

Provided that we have access to an oracle which determines whether $X \in \mathcal{I}$ for any $X \subseteq C$ in time $O(\text{poly}(n))$, Algorithm 1 has time complexity $O(\text{poly}(n))$.

$\square$

3 **2-Connectivity** *(20 points)* In this problem you are supposed to design a 2-approximation algorithm for the 2-edge connected spanning subgraph problem. Here, we are given an undirected graph $G = (V, E)$ with edge weights w_e for each $e \in E$. A 2-edge connected spanning subgraph is a set of edges $F \subseteq E$ such that every cut $\emptyset \neq S \subsetneq V$ is crossed by at least two edges, $|\delta(S) \cap F| \geq 2$. Recall that $\delta(S)$ denotes the set of edges in E with one endpoint in S and one in $V \setminus S$. In other words, one has to remove at least two edges from the graph (V, F) to disconnect it. We want to find a 2-edge connected spanning subgraph $F \subseteq E$ with minimum weight. This problem is NP-complete, so we want to come up with a good approximation algorithm. Consider the following linear programming relaxation:

$$\begin{aligned} & \min \sum_{e \in E} w_e x_e \\ \text{s.t.} \quad & \sum_{e \in \delta(S)} x_e \geq 2, \quad \forall \emptyset \neq S \subsetneq V \\ & x_e \in [0,1], \quad \forall e \in E \end{aligned} \tag{LP1}$$

A related linear program can be formulated for a weighted *directed* graph $H = (V, A)$ with a fixed vertex $r \in V$ (r can be chosen arbitrarily). It can be solved in polynomial time and has the advantage that it is *integral*, i.e. every extreme point solution has coordinates in $\{0, 1\}$.

$$\begin{aligned} & \min \sum_{a \in A} w_a y_a \\ \text{s.t.} \quad & \sum_{a \in \delta^-(S)} y_e \geq 2, \quad \forall \emptyset \neq S \subseteq V \setminus \{r\} \\ & y_a \in [0,1], \quad \forall a \in A \end{aligned} \tag{LP2}$$

where $\delta^-(S) = \{e = (u, v) \in E : u \notin S, v \in S\}$ denotes the set of edges going into S.

Prove that there exists a polynomial time 2-approximation algorithm for the 2-edge connected spanning subgraph problem. You can use the fact that (LP2) is integral and that it can be solved in polynomial time without proof.

Hint: consider the directed graph $H = (V, A)$ where each edge $e \in E$ of G is present in both directions, $(u, v) \in A$ and $(v, u) \in A$ if $\{u, v\} \in E$.

Additional hint: transform solutions from (LP1) to (LP2) and the other way round.

Solution. The algorithm constructs the graph H from the *Hint*, picks an arbitrary vertex $r \in V$ and solves (LP2) to obtain an integral optimal solution y. Then for each edge $e = \{u, v\}$ set $x_e = \max\{y_{uv}, y_{vu}\}$.
We will start by proving that x corresponds to a 2-edge connected spanning subgraph which has cost at most the cost of y in (LP2). First note that x is integral since y is integral. Next, we will show that x is a feasible solution to (LP1). Clearly $x_e \in [0, 1]$. Let $\emptyset \neq S \subseteq V$ and assume that $r \notin S$. Otherwise we can consider $V \setminus S$ since $\delta(S) = \delta(V \setminus S)$. Observe that

$$\sum_{uv \in \delta(S)} x_{uv} = \sum_{uv \in \delta(S)} \max\{y_{uv}, y_{vu}\} \geq \sum_{a \in \delta^-(S)} y_a \geq 2.$$

Thus, x is feasible for (LP1). Moreover, x has cost at most the cost of y in (LP2),

$$\sum_{e \in E} w_e x_e = \sum_{uv \in E} w_{uv} \max\{y_{uv}, y_{vu}\} \leq \sum_{uv \in E} w_{uv}(y_{uv} + y_{vu}) = \sum_{a \in A} w_a y_a.$$

It is left to show that the cost of y in (LP2) is at most twice the cost of an optimal solution $x^\star$ of (LP1). Let $x^\star$ be an optimal solution of (LP1). For each edge $uv \in E$ set $y^\star_{uv} = x^\star_{uv}$ and $y^\star_{vu} = x^\star_{uv}$. Note that $y^\star$ is feasible for (LP2). Indeed, for all $S \subseteq V \setminus \{r\}$,

$$\sum_{uv \in \delta^-(S)} y^\star_{uv} = \sum_{uv \in \delta^-(S)} x^\star_{uv} = \sum_{uv \in \delta(S)} x^\star_{uv} \geq 2.$$

The cost of $y^\star$ is equal to

$$\sum_{a \in A} y^\star_{uv} = \sum_{uv \in E} (y^\star_{uv} + y^\star_{vu}) = 2 \sum_{uv \in E} x^\star_{uv}.$$

We can conclude that our algorithm found a 2-approximate solution: x corresponds to a 2-edge connected spanning subgraph and has cost at most the cost of $y^\star$.

Remark Integrality of (LP2) follows from Corollary 53.6a. in [Schrijver A. - Combinatorial optimization, polyhedra and efficiency]. It can be solved in polynomial time since the separation oracle reduces to min-cut. $\square$

4 Integrality of the Spanning Tree LP *(31 points)* The goal of this problem is to apply linear programming techniques to the minimum spanning tree problem. Let $G = (V, E)$ be an undirected connected graph with edge weights w_e for each $e \in E$. Introduce a variable x_e for each edge $e \in E$. The minimum spanning tree linear program (ST-LP) is the following:

$$\text{(ST-LP)} \qquad \begin{aligned} \text{minimize} \quad & \sum_{e \in E} w_e x_e \\ \text{s.t.} \quad & \sum_{e \in E} x_e = |V| - 1, && (1) \\ & \sum_{e \in E[S]} x_e \leq |S| - 1, \quad \forall\, S \subseteq V, |S| \geq 2 && (2) \\ & x_e \geq 0, \quad \forall e \in E && (3) \end{aligned}$$

where $E[S]$ denotes the set of edges with both endpoints in S.

4a *(5 pts)* Let $x \in \mathbb{R}^E$ be a feasible solution for the linear program (ST-LP). Prove that $x_e \leq 1$ for each $e \in E$.

Solution. For an edge $(u, v) \in E$, the second constraint with $S = \{u, v\}$ gives $x_e \leq 1$. $\square$

4b *(10 pts)* Let $\mathcal{L}$ be any laminar family over ground set V such that each $S \in \mathcal{L}$ has size $|S| \geq 2$. Prove that $\mathcal{L}$ can consist of at most $|V| - 1$ sets, i.e. $|\mathcal{L}| \leq |V| - 1$.

Solution. Proceed by induction on $n = |V|$. If $n = 2$, at most one non-singleton set can appear, so $|\mathcal{L}| \le 1 = n - 1$. Assume the claim holds for all ground sets of size less than n, and consider a laminar family $\mathcal{L}$ on V with all sets of size at least 2.

If $\mathcal{L} = \{V\}$, then $|\mathcal{L}| = 1 \le n - 1$ and we are done. Otherwise, let $S \in \mathcal{L}$ be inclusion-wise maximal with $S \ne V$. By laminarity, every set of $\mathcal{L} \setminus \{V\}$ is either contained in S or disjoint from S. Let $\mathcal{L}_{\subseteq S} = \{X \in \mathcal{L} : X \subseteq S\}$ and $\mathcal{L}_{\cap S=\emptyset} = \{X \in \mathcal{L} : X \cap S = \emptyset\}$. Then $\mathcal{L}_{\subseteq S}$ is a laminar family on S with no singletons, so by induction $|\mathcal{L}_{\subseteq S}| \le |S| - 1$. Similarly, $\mathcal{L}_{\cap S=\emptyset}$ is laminar on $V \setminus S$ with no singletons, hence $|\mathcal{L}_{\cap S=\emptyset}| \le |V| - |S| - 1$.

If $V \notin \mathcal{L}$, then

$$|\mathcal{L}| = |\mathcal{L}_{\subseteq S}| + |\mathcal{L}_{\cap S=\emptyset}| \le (|S| - 1) + (|V| - |S| - 1) = |V| - 2 < n - 1.$$

If $V \in \mathcal{L}$, then

$$|\mathcal{L}| = 1 + |\mathcal{L}_{\subseteq S}| + |\mathcal{L}_{\cap S=\emptyset}| \le 1 + (|S| - 1) + (|V| - |S| - 1) = |V| - 1 = n - 1.$$

Thus in all cases $|\mathcal{L}| \le n - 1$, completing the induction. □

4c *(16 pts)* Assume the following fact about the linear program above (without proving it[1]).

Fact 4.1 *For every extreme point $x^* \in \mathbb{R}^E$ of the linear program (ST-LP), there exists a subset $E_0 \subseteq E$ and a laminar family[2] $\mathcal{L} \subseteq \{S \subseteq V : |S| \ge 2\}$ such that x^* is the unique solution to the following linear system:*

$$\begin{aligned} \sum_{e \in E[S]} x_e &= |S| - 1, \qquad && \forall\, S \in \mathcal{L} \\ x_e &= 0, && \forall\, e \in E_0 \end{aligned}$$

Using Fact 4.1 and the claims in **4a** and **4b**, prove that every extreme point $x^* \in \mathbb{R}^E$ of the linear program (ST-LP) satisfies $x^*_e \in \{0, 1\}$ for each $e \in E$, i.e. (ST-LP) is integral.

Hint: if a linear system with t equations in m variables has a unique solution, then $t \ge m$.

Solution. By Fact 4.1 there exists $E_0 \subseteq E$ and a laminar subfamily $\mathcal{L} \subseteq \{S \subseteq V : |S| \ge 2\}$ such that x^* is the unique solution to the linear system

$$\begin{aligned} \sum_{e \in E[S]} x_e &= |S| - 1 \qquad && \forall\, S \in \mathcal{L} \\ x_e &= 0 && \forall\, e \in E_0 \end{aligned}$$

Hence, the number of equations must be at least $|E|$. Since the number of equations is $|\mathcal{L}| + |E_0|$ and $|\mathcal{L}| \le |V| - 1$ by Question 4.b, we get that $|E| \le |V| - 1 + |E_0|$. We know by Question 4.a that $0 \le x^*_e \le 1$. Also, we have $x^*_e = 0$ for all $e \in E_0$, so all those coordinates are integral. Hence, if there is an edge $e \in E \setminus E_0$ with $x^*_e < 1$, we have that

$$\sum_{e \in E} x^*_e = \sum_{e \in E \setminus E_0} x^*_e < |E \setminus E_0| = |E| - |E_0| \le |V| - 1\,,$$

which contradicts the first constraint in the LP. Therefore, we conclude that x^* is integral. □

[1] One can prove Fact 4.1 by following the same approach as in the third problem ("minimum transportation") of Problem Set 1. You are not asked to prove Fact 4.1.

[2] Recall that a collection of sets $\mathcal{L}$ is called laminar if for all $S, T \in \mathcal{L}$ either $S \subseteq T$, or $T \subseteq S$, or $S \cap T = \emptyset$.