CS182 Fall 2022
Midterm Exam

Name: ______________________________

Harvard ID: _________________________

Name: ________________________________

Question 1: Linear and Integer Programming

1. **[10 pts]** Consider an integer program $\mathcal{P}$ with binary variables in $\{0,1\}$, and let $\mathcal{P}'$ be its LP relaxation, which is identical except that the variables are constrained to be in $[0,1]$ instead of $\{0,1\}$. Assuming $\mathcal{P}$ is a maximization problem, what is the relation between the optimal solutions of $\mathcal{P}$ and $\mathcal{P}'$? Explain your answer.

   1. Let $opt(\cdot)$ denote the optimal solution value of a problem. We have that $opt(\mathcal{P}') \geq opt(\mathcal{P})$ because any solution to the integer program $\mathcal{P}$ is a feasible solution to $\mathcal{P}'$, so the optimal solution in $\mathcal{P}'$ can only be at least as good.

2. **[15 pts]** The *set cover* problem is defined as follows. There is a set $U = 1, \ldots, n$ of n elements, a set $\mathcal{F} = \{F_1, \ldots, F_m\}$ of subsets of U (i.e., $F_i \subseteq U$ for all $i = 1, \ldots, m$), and an integer k. The goal is to choose k subsets from the collection $\mathcal{F}$ whose union is as large as possible. In other words, the goal is to choose $S \subseteq \{1, \ldots, m\}$ such that $|S| = k$ to maximize $|\bigcup_{j \in S} F_j|$.

   Formulate the set cover problem as an integer program.

   Hint: Use binary variables $x_1, \ldots, x_n$ such that $x_i = 1$ if and only if element i is covered, and binary variables $y_1, \ldots, y_m$ such that $y_j = 1$ if and only if subset F_j is included in the solution S. The main challenge is to write a linear constraint that allows x_i to be 1 if and only if i is covered using the appropriate y_j variables.

2. Define the variables as in the hint. Furthermore, define binary constants z_{ij}, $1 \leq i \leq n$, $1 \leq j \leq m$, such that $z_{ij} = 1$ if and only if F_j contains element i. We have the following constraints:

- $x_i \in \{0,1\}, \quad \forall 1 \leq i \leq n.$
- $y_i \in \{0,1\}, \quad \forall 1 \leq j \leq m.$
- $x_i \leq \sum_{j=1}^{m} z_{ij} y_j, \quad \forall 1 \leq i \leq n.$
- $\sum_{j=1}^{m} y_j = k.$

and our objective is:

- $\max \sum_{i=1}^{n} x_i$

The third and fourth constraint ensures that $x_i = 1$ if and only if at least one of the y_j's corresponding to an F_j that contains element i is included in the solution.

Name: ________________________________

Question 2: Game Theory and Convex Optimization

1. **[10 pts]** Define (using mathematical notation) the following terms: convex function, convex set, and convex optimization problem.

1. A convex function f is any function $f : X \to \mathbb{R}$ such that $\forall \theta \in [0, 1]$, $\forall x, y \in X$, we have $f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$.

A set C is convex if and only if $\forall x, y \in C$, $\forall \theta \in [0, 1]$, $\theta x + (1 - \theta)y \in C$.

A convex optimization problem is the minimization of a convex function on a domain that is a convex set.

Name: ______________________________

2. **[15 pts]** Consider a 2-player game in normal form, and denote the strategy set of each player by S. Let $B_1(x_2)$ denote the set of (possibly mixed) best response strategies of player 1 to the (possibly mixed) strategy x_2 of player 2. For convenience, let us fix some mixed strategy $x_2^\star$ for player 2, and denote $\alpha = u_1(x_1, x_2^\star)$ for all $x_1 \in B_1(x_2^\star)$, that is, α is the maximum utility player 1 can achieve against $x_2^\star$.

   Show that $x_1 \in B_1(x_2^\star)$ *if and only if* every pure strategy $s \in S$ in the support of x_1 (i.e., every pure strategy s such that $x_1(s) > 0$) is itself in $B_1(x_2^\star)$ (i.e., $u_1(s, x_2^\star) = \alpha$).

   Note: Do not forget to show both directions.

2. ($\Longrightarrow$) Suppose that $x_1 \in B_1(x_2^\star)$. Suppose for contradiction that $\exists s^\star \in S$, but $s^\star \notin B_1(x_2^\star)$. Let $t^\star \in B_1(x_2^\star)$ be a pure strategy. Then we have:

$$\begin{aligned} u_1(x_1, x_2^\star) &= \alpha \\ &= \sum_{s \in S} x_1(s) u_1(s, x_2^\star) \\ &< x_1(s^\star) u_1(t^\star, x_2^\star) + \sum_{s \in S, s \neq s^\star} x_1(s) u_1(s, x_2^\star) \end{aligned}$$

so it is strictly better for x_1 to increase its probability of playing $t^\star$ by $x_1(s^\star)$ while decreasing the probability of playing $s^\star$ by $x_1(s^\star)$, thus contradicting that $x_1 \in B_1(x_2^\star)$.

($\Longleftarrow$) If every pure strategy $s \in S$ in the support of x_1 is in $B_1(x_2^\star)$, then $u_1(s, x_2^\star) = \alpha$ for all $s \in S$, so $u_1(x_1, x_2^\star) = \sum_{s \in S} x_1(s) u_1(s, x_2^\star) = \alpha$, so $x_1 \in B_1(x_2^\star)$.

Name: ______________________________

3. **[BONUS 5 pts]** Assuming that part 2 has been established, show that it follows that the set $B_1(x_2^\star)$ is convex.

   Note: Each x_1 in this set is a vector of probabilities for pure strategies $s \in S$.

   Hint: The length of the proof should be 2-4 lines.

3. Suppose that $\ell_1, \ell_2 \in B_1(x_2^\star)$ (possibly mixed strategies). Then for all $\theta \in [0,1]$, we have $\theta\ell_1 + (1-\theta)\ell_2 = (\theta\ell_1(s_1) + (1-\theta)\ell_2(s_1), \ldots, \theta\ell_1(s_n) + (1-\theta)\ell_2(s_n))$, which is also a valid vector of probabilities because

$$\sum_{i=1}^{n} \theta\ell_1(s_i) + (1-\theta)\ell_2(s_i) = 1.$$

Moreover, because both ℓ_1 and ℓ_2 are supported by pure strategies in $B_1(x_2^\star)$, we must have that this strategy is in $B_1(x_2^\star)$ by the previous problem.

Name: ____________________________________

Question 3: Informed Search

1. **[10 pts]** Define (using mathematical notation) the following terms: *admissible* heuristic, *consistent* heuristic.

1. An admissible heuristic h satisfies $h(x) \leq h^*(x)$ for all states x, where h^* is the optimal distance to the goal.

A consistent heuristic h satisfies $h(x) \leq c(x, y) + h(y)$ for all states x, y where $c(x, y)$ is the cost of the optimal (cheapest) path from state x to y.

Name: ________________________________

2. **[15 pts]** Recall that in an 8-puzzle the goal state is:

1	2	3
4	5	6
7	8	

Consider the following (weird) heuristic h. On the following specific state $s^\star$, $h(s^\star)$ is the sum of Manhattan distances between the tiles and their positions in the goal state, i.e., $h(s^\star) = 13$:

8	2	4
3	1	5
7		6

On every other state s, $h(s)$ is equal to the number of misplaced tiles.

Answer the following two questions:

(a) Is h admissible?

(b) Is h consistent?

If your answer is "yes," provide a proof (it can be very short), and if it is "no," provide an explicit counterexample.

2.

(a) h is admissible. For all $s \neq s^\star$, we have $h(s) \leq h^*(s)$ because each swap moves a tile closer to the goal state by at most 1.

(b) h is not consistent. Consider moving from $s^\star$ to the state s where we swap the blank tile and the tile with the 6. The resulting state has $h(s) = 6$ (6 misplaced tiles), while $h(s^\star) = 13$, so $h(s^\star) > c(s^\star, s) + h(s)$, which violates consistency.

Name: ______________________________

Question 4: Constraint Satisfaction Problems

1. **[10 pts]** Describe in words what *forward checking* does.

   1. Forward checking is a way of using inference within searching to eliminate obvious violations and find feasible solutions in constraint satisfaction problems. In particular, we can enforce arc consistency.

2. **[15 points]** We say that a binary CSP (which has only binary constraints) with variables $X_1, \ldots, X_n$ is *k-consistent* if and only if for every subset of k variables $X_1, \ldots, X_k$, any legal assignment for $X_1, \ldots, X_{k-1}$ can be extended to a legal assignment for $X_1, \ldots, X_k$. More formally, for every assignment for $X_1, \ldots, X_{k-1}$ that satisfies the binary constraints on these variables, there exists a value in D_k that satisfies all the binary constraints of the form (X_j, X_k) for $j \in [k-1]$. (Note that arc-consistency is equivalent to 2-consistency.) A binary CSP is *strongly k-consistent* if and only if it is k'-consistent for all $k' \leq k$.

   Suppose that a given binary CSP with n variables is strongly n-consistent, and suppose that this CSP is solved using "vanilla" backtracking search (without any heuristics like forward checking). Give an upper bound on the number of times the search would have to backtrack, i.e., the number of times it would have to undo an assignment to a variable. Explain your answer.

   Note: Assume that backtracking search only assigns a value to a variable when it is consistent with the existing partial assignment.

   Hint: The length of the solution should be 2-3 lines.

   2. The search would never have to backtrack, that is, the upper bound is 0. This is because for any partial assignment to the variables, a valid value can always be assigned to the next variable.

Name: ______________________________

Additional Space / Scratch Paper