(1) (25 points) Multiple Choice Questions: Select all the answers that you think are correct. These questions may have one or more correct answers. You should circle all that apply.
Solution:

1. *(iv)*
2. *(iv)*
3. *(i)(iii)(v)*
4. *(ii)(iii)(v)(vi)(vii)*
5. *(ii)*

(2) (**15 points**) **Optimization:** Problem Formation: Suppose that $f(x) = h(g_1(x), g_2(x), \cdots, g_k(x))$, were $h : \mathcal{R}^k \to R$ is convex, and $g_i : \mathcal{R}^n \to \mathcal{R}$. Suppose that for each i,

(a) (**5 points**) h is nondecreasing in the $i-$ th argument, and g_i is convex, is f convex? Why?

(b) (**5 points**) h is nonincreasing in the $i-$ th argument, and g_i is concave, is f convex? Why?

(c) (**5 points**) g_i is a linear function, is f convex? Why?

Solution. Fix x, y, and $\theta \in [0,1]$, and let $z = \theta x + (1-\theta)y$. Let's re-arrange the indexes so that g_i is affine for $i = 1, \ldots, p$, g_i is convex for $i = p+1, \ldots, q$, and g_i is concave for $i = q+1, \ldots, k$. Therefore we have

$$\begin{aligned} g_i(z) &= \theta g_i(x) + (1-\theta) g_i(y), \qquad i = 1, \ldots, p, \\ g_i(z) &\le \theta g_i(x) + (1-\theta) g_i(y), \qquad i = p+1, \ldots, q, \\ g_i(z) &\ge \theta g_i(x) + (1-\theta) g_i(y), \qquad i = q+1, \ldots, k. \end{aligned}$$

We then have

$$\begin{aligned} f(z) &= h(g_1(z), g_2(z), \ldots, g_k(z)) \\ &\le h(\theta g_1(x) + (1-\theta) g_1(y), \ldots, \theta\, g_k(x) + (1-\theta) g_k(y)) \\ &\le \theta h(g_1(x), \ldots, g_k(x)) + (1-\theta) h(g_1(y), \ldots, g_k(y)) \\ &= \theta f(x) + (1-\theta) f(y). \end{aligned}$$

The second line holds since, for $i = p+1, \ldots, q$, we have increased the ith argument of h, which is (by assumption) nondecreasing in the ith argument, and for $i = q+1, \ldots, k$, we have decreased the ith argument, and h is nonincreasing in these arguments. The third line follows from convexity of h.

(3) (**15 points**) **Linear Programming:**

(a) (**5 points**) If x and y satisfy the conditions that:

$$\begin{cases} 2x + y - 2 \geq 0 \\ x - 2y + 4 \geq 0 \\ 3x - y - 3 \leq 0 \end{cases}$$

Then what is the largest and smallest value of $z = x^2 + y^2$?

(b) (**10 points**) Let $A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$, $C^T = (3, 2, 1, 2, 3)$ and $b = \begin{pmatrix} 2 \\ 7 \end{pmatrix}$.

Consider the corresponding linear programming problem in standard form:

minimize $C^T x$ subject to $Ax = b, x \geq 0$

(i) Determine an optimal solution and the optimal cost.

(ii) Is the optimal solution unique?

(iii) How many basic solutions are there?

(iv) How many basic feasible solutions are there?

(a) One optimal solution is $x^* = (0, 0, 1, 1, 0)^T$ and the optimal cost is $c^T x^* = 3$.

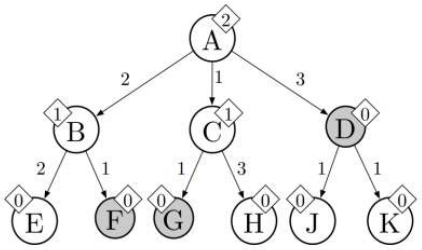
(b) No, it is not unique. Another optimal solution is $x^{**} = (0, 0, 3/2, 0, 1/2)$.

(c) Any two columns of A are linearly independent, giving $\binom{5}{2} = 10$ basic solutions.

(d) The feasible bases are obtained by combing any of the first three columns with either of the last two columns. These are the pairs of columns that have b in their positive span. Hence there are 6 basic feasible solutions.

(4) (**20 points**) **Search algorithms**

Consider the state search problem shown below. A is the start state and the shaded states are goal states. Arrows are possible state transitions, and numbers by the arrows are costs of each action. Numbers in diamond shape are heuristic values that are used to estimate the optimal cost from that node to the goal.



For each of the following search algorithms, please write down the nodes removed from the fringe and the path it returned. Assumed that the data structure implementations and successor state ordering are all such that ties are broken alphabetically. For example, a partial plane $S \to X \to A$ would be expanded before $S \to X \to B$.

(a) (**4 points**) Depth First Search (ignores costs)
Nodes removed from fringe:
Path returned:

(b) (**4 points**) Breadth First Search (ignores costs)
Nodes removed from fringe:
Path returned:

(c) (**4 points**) Uniform Cost Search
Nodes removed from fringe:
Path returned:

(d) (**4 points**) Greedy Search
Nodes removed from fringe:
Path returned:

(e) (**4 points**) A^* Search
Nodes removed from fringe:
Path returned:

(a) DFS:
Nodes removed from fringe: A, B, E, F
Path returned: A, B, F

(b) BFS
Nodes removed from fringe: A, B, C, D
Path returned: A, D

(c) UCS
Nodes removed from fringe: A, C, B, G
Path returned: A, C, G

(d) Greedy Search

Nodes removed from fringe: A, D
Path returned: A, D

(e) A*
Nodes removed from fringe: A, C, G
Path returned: A, C, G

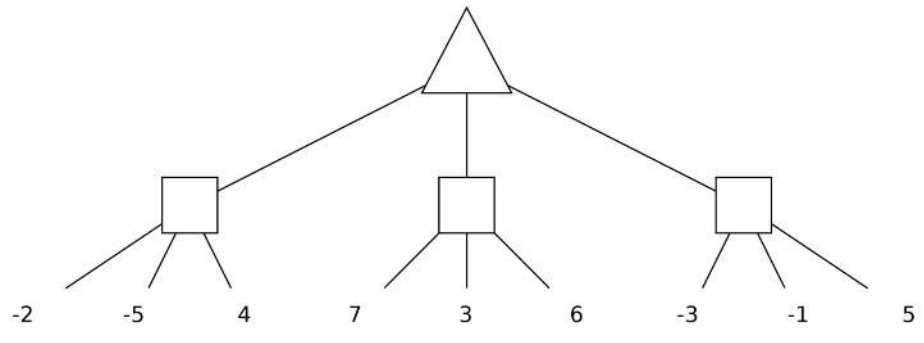
(5) (**15 points**) **CSP Formulation:**
Nodes S, B, M, D, P, G are lining up next to each other with numbered positions 1, 2, 3, 4, 5, 6, where 1 neighbors 2, 2 neighbors 1 and 3, 3 neighbors 2 and 4, 4 neighbors 3 and 5, 5 neighbors 4 and 6, and 6 neighbors 5. Each one of them takes up exactly one spot. B needs to be next to M one one side and D on the other side. P needs to be next to the G. S needs to be at 1 or 2. Formulate this problem as a CSP: list the variables, their domains, and the constraints. Encode unary constraints as a constraint rather than pruning the domain. (You do not need to solve the problem, just provide variables, domains and implicit constraints).
(a) (**5 points**) Variables:
(b) (**5 points**) Domains:
(c) (**5 points**) Constraints:

(a) Variables: S, B, M, D, P, G
(b) Domains: {1, 2, 3, 4, 5, 6} for all variables
(c) Constraints: alldiff(S, B, M, D, P, G), |B – M | = 1, |B - P| = 1, |P – G| = 1, S∈{1, 2}

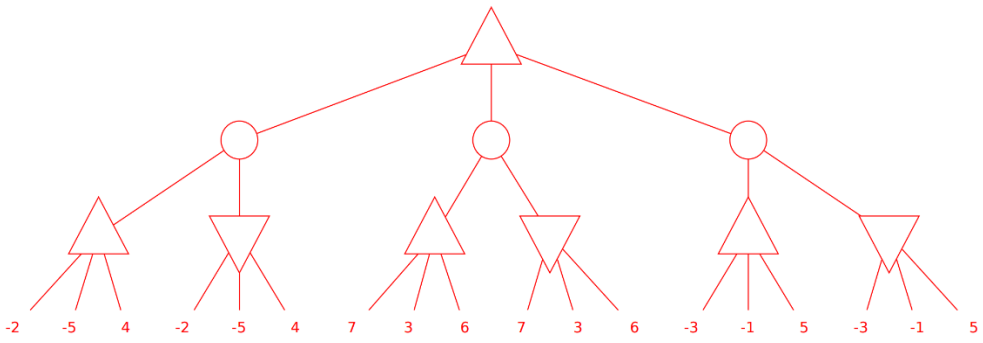
(6) **(10 points) Zero-sum game**

Consider a zero-sum game with two players, one's goal is to maximize agent and one's goal is to minimize agent. In this game, the ordering of moves is no longer deterministic.Each turn, a coin is flipped in order to determine which agent gets to make a move during that time step.

Consider the game tree below which playing for two turns. It is currently the move from the maximizer, so the top node is a max node. As we don't know which agent is going to play on the next turn, we've replaced those nodes with boxes. Draw a new game tree that consists of only the traditional min, max, and expecti-nodes that models this situation. Then, fill in the values in each boxes and perform α-β pruning.



Hint for question (6):



After transformation, then fill in the values and conduct the pruning.