CSE 4102
FL 25

# Practice Exam

Exam1

- You have approximately 80 minutes.
- Mark your answers ON THE EXERCISE ITSELF. If you are not sure of your answer you may wish to provide a *brief* explanation. All short answer sections can be successfully answered in a few sentences AT MOST.
- For True/False questions, please *circle* your answer.

| | |
|---|---|
| First name | |
| Last name | |
| WUSTL ID | |

**For staff use only:**

| | | |
|---|---|---|
| Q1. | True/False Questions | /15 |
| Q2. | Search Algorithms | /26 |
| Q3. | Adversarial Search | /17 |
| Q4. | Constraint Satisfaction Problems (CSPs) | /22 |
| | Total | /80 |

# Q1. [15 pts] True/False Questions

Each question is worth 1 point. Leaving a question blank is worth 0 points.

**(a)** Search

**(i)** [1 pt] [*true* or *false*] With the same tie-breaking criteria, iterative-deepening depth-first search and breadth-first search expand exactly the same number of nodes.

**(ii)** [1 pt] [*true* or *false*] With the same tie-breaking criteria, uniform-cost search will always expand more nodes than A* search.

**(iii)** [1 pt] [*true* or *false*] The heuristic $h(n) = 0$ is consistent for every search problem with non-negative costs.

**(iv)** [1 pt] [*true* or *false*] Greedy best-first search, using a perfect heuristic $h(n) = h^*(n)$ that always returns the true cost to the goal is optimal.

**(v)** [1 pt] [*true* or *false*] A* search with an admissible but inconsistent heuristic is correct and complete without re-generating and re-expanding nodes.

**(b)** Adversarial Search

**(i)** [1 pt] [*true* or *false*] With identical tie-breaking strategies, minimax search is guaranteed to return the exact same solution as alpha-beta search.

**(ii)** [1 pt] [*true* or *false*] When playing against a random agent, minimax will compute an optimal solution.

**(iii)** [1 pt] [*true* or *false*] Minimax is a natural choice for modeling poker.

**(iv)** [1 pt] [*true* or *false*] An evaluation function for expectimax must always return positive values.

**(v)** [1 pt] [*true* or *false*] Expectimax solutions are gauranteed to always perform better on average (i.e., it gets a higher utility when playing the same game a very large number of times) than minimax solutions.

**(c)** Constraint Satisfaction Problems

**(i)** [1 pt] [*true* or *false*] The Minimum Remaining Value (MRV) heuristic is a variable-ordering heuristic.

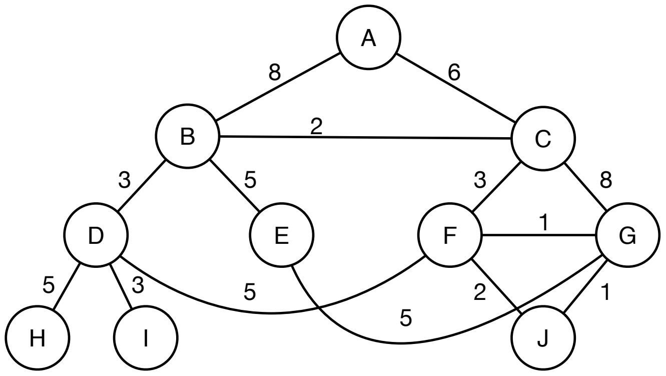
**(ii)** [1 pt] [*true* or *false*] Breadth-first search always expands at least as many nodes as depth-first search when solving CSPs with the same variable- and value-ordering heuristics.

**(iii)** [1 pt] [*true* or *false*] Depth-first search always expands at least as many nodes as a search algorithm that chooses random fringe nodes to expand when solving CSPs with the same variable- and value-ordering heuristic.

**(iv)** [1 pt] [*true* or *false*] After running arc consistency on a CSP, it is impossible for backtracking to prune a value during forward checking.

**(v)** [1 pt] [*true* or *false*] Hill climbing with a random initial assignment is guaranteed to find a solution to a CSP, if one exists.

## Q2. [26 pts] Search Algorithms



The question here refer to the graph above, where the start state is $A$ and the goal state is $G$. The number on an edge corresponds to the cost of traversing that edge.

Additionally, assume that we have the following heuristic values:

| State | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| Heuristic value | 8 | 4 | 1 | 4 | 2 | 3 | 0 | 4 | 5 | 2 |

Assume that each algorithm re-generates states that are not yet expanded does not re-expand states, breaks ties in lexicographical ordering, and terminates after expanding the goal state.
*Note: These assumptions may differ from the operations of some of the algorithms in the textbook.*

**(a)** [7 pts] What is the order of state expansions if you used Breadth-First Search (BFS)?
(If state $A$ is expanded before state $B$, which is expanded before state $C$, then write "$A$, $B$, $C$".)

What path would the algorithm return?
(If the path is from state $A$ to state $B$ to state $C$, then write "$A \to B \to C$".)
What is the cost of the path returned by the algorithm?

**(b)** [7 pts] What is the order of state expansions if you used A* with the heuristics provided in the above table? What path would the algorithm return? What is the cost of the path?

Consider a search problem with unit edge costs (i.e., the cost of each edge is 1) and two consistent heuristic functions $h_1$ and $h_2$. Recall that a heuristic function $h$ is consistent if it satisfies the triangle inequality:

$$h(n) \leq c(n, n') + h(n')$$

where $c(n, n')$ is the optimal cost between nodes $n$ and $n'$, and also satisfies $h(goal) = 0$. Let $h_3(n) = \max(h_1(n), h_2(n))$ denote a new heuristic function.

**(c)** [6 pts] Is the heuristic function $h_3$ admissible? If it is, prove it. If it is not, give a counter-example.

**(d)** [6 pts] Is the heuristic function $h_3$ consistent? If it is, prove it. If it is not, give a counter-example.

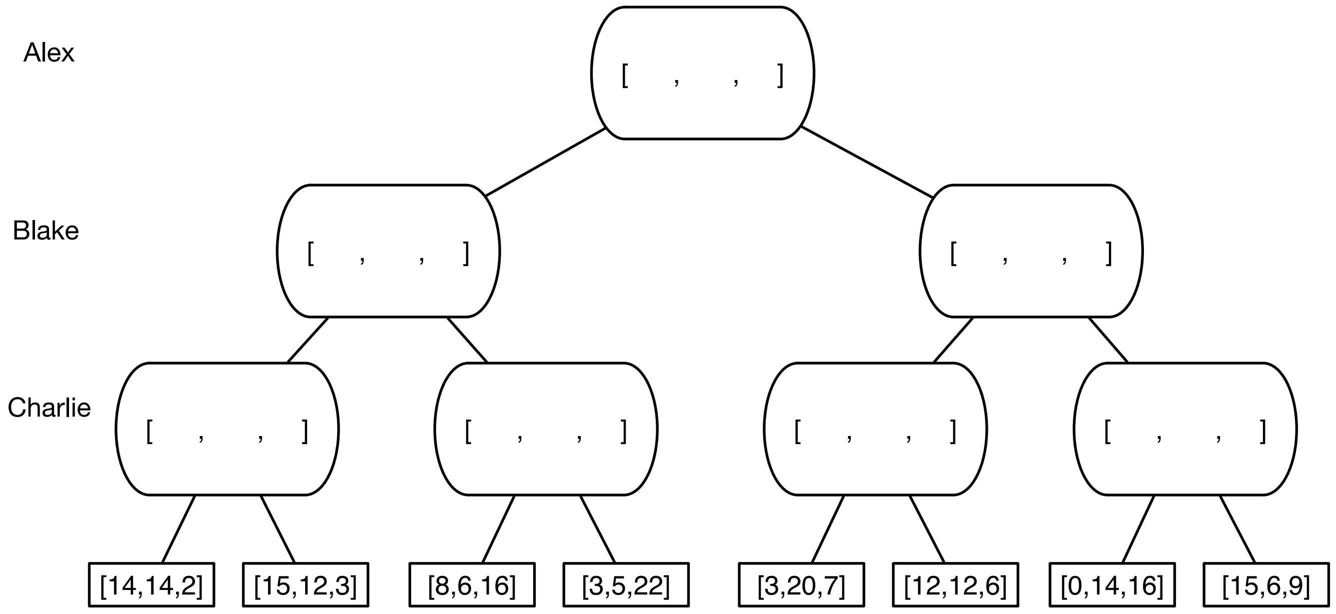
## Q3. [17 pts] Adversarial Search

Alex, Blake, and Charlie are three friends who are trying to decide how to strategically divide the 30 Halloween candies that they have collected. They finally decided to use a candy shuffler, which takes in a set number of candies and groups them into three batches, one for each player. The candy shuffler has three levers (with either "up" or "down" position), and the three levers combined determine how the candies are distributed among the three players.

After putting their 30 cookies into the shuffler, Alex will pull the first lever, followed by Blake who will pull the second lever, and finally Charlie will pull the last lever. The utility triples (= outcomes) after all three levers are pulled are shown in the game tree below, where left branches correspond to "up" positions and right branches correspond to "down" positions. For example, if all three players pulled their levers to the "up" positions, then the utility triple is $[14, 14, 2]$, which means that Alex gets 14 candies, Blake gets 14 candies, and Charlie gets 2 candies.

Assume all players want to maximize their own number of candies. Further, no candies are lost in the process. So, the sum of candies of all three players must equal 30.

**(a)** [14 pts] Fill in the utility triples in all the nodes in the game tree below.



**(b)** [3 pts] What is the optimal action of each player? Highlight the corresponding branch in the game tree above.

## Q4. [22 pts] Constraint Satisfaction Problems (CSPs)

We have five planes: A, B, C, D, and E and two runways: international and domestic. We would like to schedule a time slot and runway for each aircraft to either land or take off. We have four time slots: $\{1, 2, 3, 4\}$ for each runway, during which we can schedule a landing or take off of a plane. We must find an assignment that meets the following constraints:

- Plane B has lost an engine and must land in time slot 1.
- Plane D can only arrive at the airport to land during or after time slot 3.
- Plane A is running low on fuel but can last until at most time slot 2.
- Plane D must land before plane C takes off, because some passengers must transfer from D to C.
- No two aircrafts can reserve the same time slot for the same runway.

**(a)** [7 pts] Model this problem as a CSP. Clearly state the variables, domains, and constraints for your CSP.

For the following subparts, we add the following two constraints:

- Planes A, B, and C cater to international flights and can only use the international runway
- Planes D and E cater to domestic flights and can only use the domestic runway.

**(b)** [5 pts] With the addition of the two constraints above, we reformulate the CSP. Assume, the variables are {A, B, C, D, E} and domains are {1, 2, 3, 4}. Clearly *draw* the constraint graph (with nodes/variables and their edges/constraints) for this problem given the original constraints and the two added ones.

**(c)** [4 pts] What are the domains of the variables after enforcing arc-consistency? Begin by enforcing unary constraints, for instance, plane B must land in time slot 1. (Cross out values that are no longer in the domain, you may use the table below.)

| $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|
| 1 2 3 4 | 1 | 1 2 3 4 | 1 2 3 4 | 1 2 3 4 |

**(d)** [6 pts] Arc-consistency can be rather expensive to enforce, and we believe that we can obtain faster solutions using only forward-checking on our variable assignments. Using the Minimum Remaining Values heuristic, perform backtracking search on the graph, breaking ties by picking lower values and characters first. List the (variable, value) pairs in the order they occur (including the assignments that are reverted upon reaching a dead end). Enforce unary constraints before starting the search, you don't have to use the table below.

| $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|
| 1 2 3 4 | 1 2 3 4 | 1 2 3 4 | 1 2 3 4 | 1 2 3 4 |