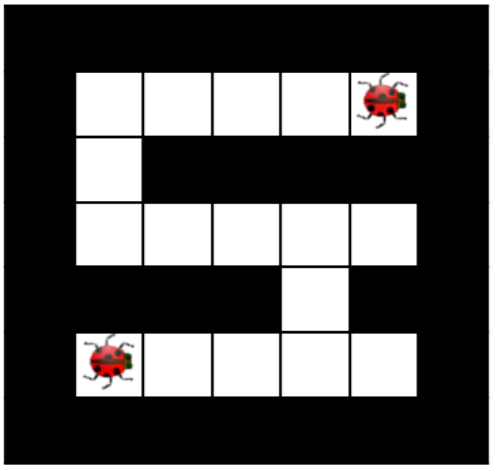
1 State Space



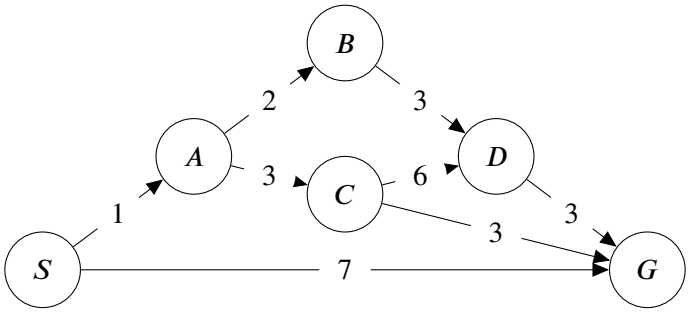
You now control a pair of long lost bug friends. You know the maze, but you do not have any information about which square each bug starts in. (Their starting positions are not known.) The bugs *cannot* jump onto walls. You want to help the bugs reunite. You must pose a search problem whose solution is an all-purpose sequence of actions such that, after executing those actions, both bugs will be on the same square, regardless of their initial positions. Any square will do, as the bugs have no goal in mind other than to see each other once again. Both bugs execute the actions mindlessly and do not know whether their moves succeed; if they use an action which would move them in a blocked direction, they will stay where they are. Both bugs can move in each time step. Every time step that passes has a cost of one.

Note that this figure of the bugs in the maze is just one possible example of the problem; the dimensions of the maze and walls will not in general be as they appear here.
Hint: There is no need to separately keep track of the bugs since their starting positions are not known. Remember, we know all the positions in the maze.

1. Give a *minimal* state representation for the above search problem. You may use variables M and N to represent the dimensions of the maze.

2. Give the size of the state space for this search problem in terms of M and N.

2 General Search



Answer the following questions about the search problem shown above. Break any ties alphabetically. S is the start state and G is the goal state. For the questions that ask for a path, please give your answers in the form '$S - A - D - G$.'

1. What path would breadth-first graph search return for this search problem?

2. What path would uniform cost graph search return for this search problem?

3. What path would depth-first graph search return for this search problem?

4. What path would A* graph search, using a consistent heuristic, return for this search problem?

5. Consider the heuristics for this problem shown in the table below. **Justify your answers for the following questions in 1-2 sentences.**

State	h_1	h_2
S	4	3
A	2	2
B	5	6
C	2	1
D	3	3
G	0	0

(a) Is h_1 admissible? **Yes** **No**

(b) Is h_1 consistent? **Yes** **No**

(c) Is h_2 admissible? **Yes** **No**

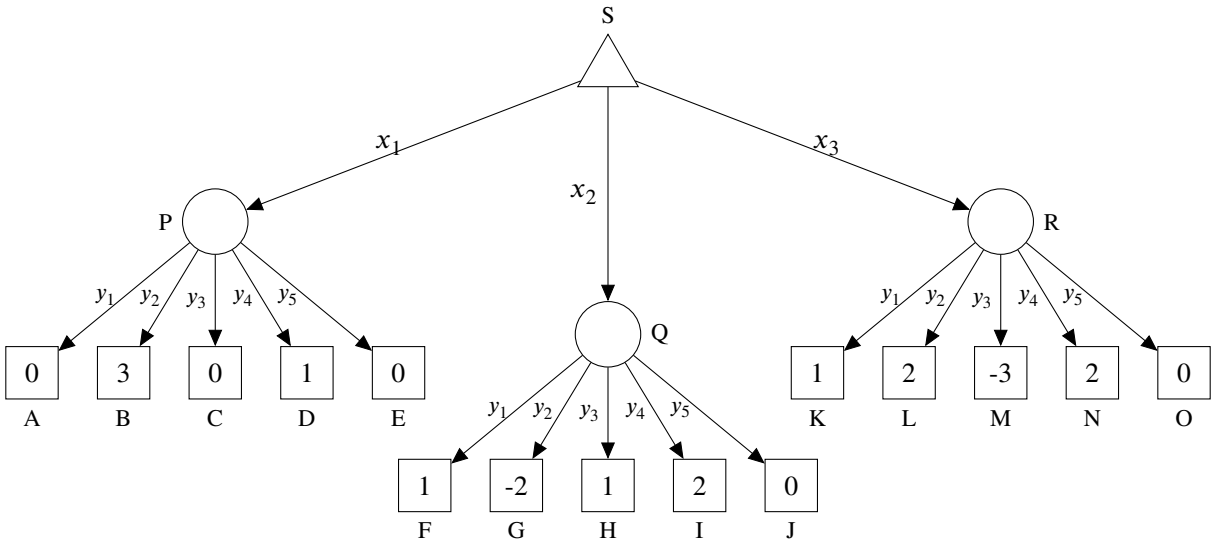
(d) Is h_2 consistent? **Yes** **No**

(e) Is $max(h_1,h_2)$ admissible? **Yes** **No**

3 Multi-agent Search

Alyssa P. Hacker and Ben Bitdiddle are bidding in an auction at Stanley University for a bike. Alyssa will either bid x_1, x_2, or x_3 for the bike. She knows that Ben will bid y_1, y_2, y_3, y_4, or y_5, but she does not know which.

1. Alyssa wants to maximize her payoff given by the expectimax tree below. The leaf nodes show Alyssa's payoff. The nodes are labeled by letters, and the edges are labeled by the bid values x_i and y_i. The maximization node S represents Alyssa, and the branches below it represent each of her bids: x_1, x_2, x_3. The chance nodes P, Q, R represent Ben, and the branches below them represent each of his bids: y_1, y_2, y_3, y_4, y_5.



(a) Suppose that Alyssa believes that Ben would bid any bid with equal probability. What are the values of the chance (circle) and maximization (triangle) nodes?

i. Node P _____

ii. Node Q _____

iii. Node R _____

iv. Node S _____

(b) Based on the information from the above tree, how much should Alyssa bid for the bike?

○ x_1 ○ x_2 ○ x_3

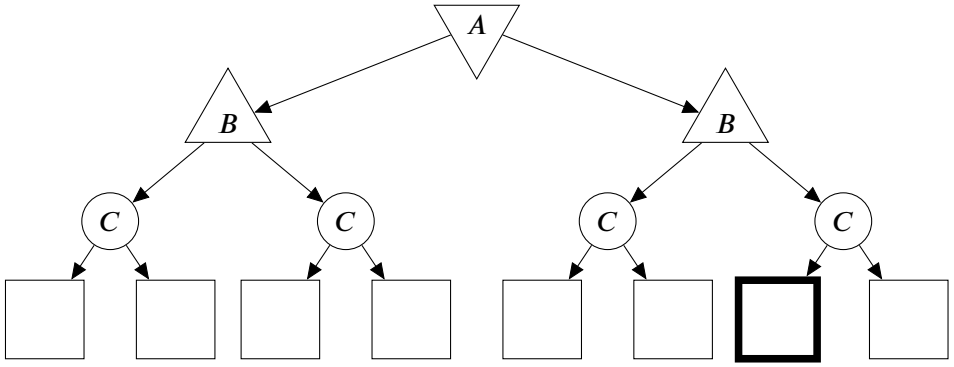
4 Alpha-Beta

In this question, player A is a minimizer, player B is a maximizer, and C represents a chance node. All children of a chance node are equally likely. Consider a game tree with players A, B, and C. In lecture, we considered how to prune a minimax game tree - in this question, you will consider how to prune an expinimax game tree (like a minimax game tree but with chance nodes). Assume that the children of a node are visited in left-to-right order.

For each of the following game trees, write "possible" if there are terminal values of the leaf nodes such that the bolded node can be pruned (it doesn't matter if you prune more nodes) or write "not possible" if no such assignment exists.

Important: The α-β pruning algorithm does not deal with chance nodes. Instead, for a node n, consider all the values seen so far, and determine whether you can know, *without looking at the node*, that the value of the node will not affect the value at the top of the tree. If that is the case, then n can be pruned.

1.



2.