CS444 Search Exercise

1. **DFS Exercise**

   - Open the AISpace search tool.
   - Load the "Extended Tree Graph" problem:
     `File` → `Load Sample Problem` → `Extended Tree Graph`
   - Open the "Solve" tab.
   - Select "Depth First Search" from the "Search Options" menu.
   - Step through the search using the "Fine Step" button.
     - Can you anticipate which node will be chosen from the frontier?
     - What is the largest number of nodes that end up on the frontier at any given time?

2. **BFS Exercise**

   - Repeat the previous steps using breadth-first search.
   - Which algorithm used more space for the frontier?

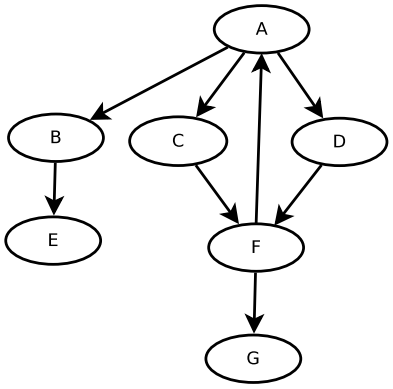
3. **DFS and Loops**

   - Open the "Create" tab.
   - Add an edge from $N4$ to S.
   - Run a depth-first search on the resulting graph.
   - Enable multi-path pruning and repeat the search.

4. **DFS vs BFS**

   - Remove the edge from $N4$ to S.
   - Add an edge from $N20$ to $N2$.
   - Search using DFS and BFS. Do they find the same path?

The next three questions refer to the following state transition graph:



For each question assume that A is the start state, E is the goal state, and each action has a cost of 1. Complete each search using multiple-path pruning.

5. **DFS by Hand**

Fill in the following table by tracing the execution of a depth-first search of the graph above. (Assume that when a node is expanded, its children are added to the frontier in alphabetical order.)

Iteration #	Frontier	Explored
1.	*A*	-
2.		
3.		
4.		
5.		
6.		
7.		
8.		
9.		
10.		

6. **BFS by Hand**

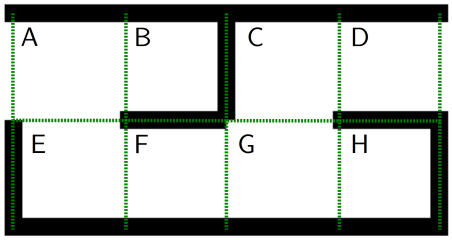
   Fill in the following table by tracing the execution of a breadth-first search of the graph above. (Assume that when a node is expanded, its children are added to the frontier in alphabetical order.)

Iteration #	Frontier	Explored
1.	A	-
2.		
3.		
4.		
5.		
6.		
7.		
8.		
9.		
10.		

7. It turns out that both searches considered above find the optimal solution to this problem. Modify the graph to create a new problem that causes DFS to find a non-optimal solution. Will BFS find the optimal solution for this new problem?

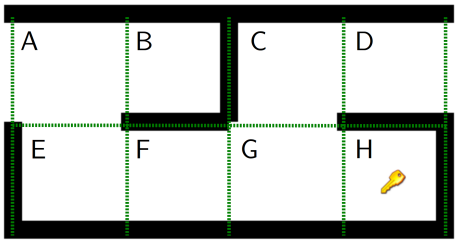
8. **Dungeon Adventure #1**

   Use the AISpace search tool to create a graph representation of the following maze navigation problem:



   In this maze it is possible for the agent to move between any two adjacent rooms unless they are blocked by a wall. Assume that A is the start state and D is the end state. Once you have completed your graph, use BFS or DFS to find a solution.

9. **Dungeon Adventure #2** Modify your graph from the previous exercise to represent the fact that there is a key at position H that must be collected before the agent can exit the maze:



   Assume that the key cannot be dropped once it is picked up.

   Use DFS or BFS to find a solution.

   How many states would this problem have if the key could be dropped in any room?