

hw1_search_q10_early_goal_checking_graph_search

Question 10: Early Goal Checking Graph Search

2/2 points (ungraded)

Recall from lecture the general algorithm for GRAPH-SEARCH reproduced below.

```
function GRAPH-SEARCH(problem, fringe, strategy) return a solution, or failure
  closed ← an empty set
  fringe ← INSERT(MAKE-NODE(INITIAL-STATE[problem]), fringe)
  loop do
    if fringe is empty then return failure
    node ← REMOVE-FRONT(fringe, strategy)
    if GOAL-TEST(problem, STATE[node]) then return node
    if STATE[node] is not in closed then
      add STATE[node] to closed
      for child-node in EXPAND(STATE[node], problem) do
        fringe ← INSERT(child-node, fringe)
      end
    end
  end
```

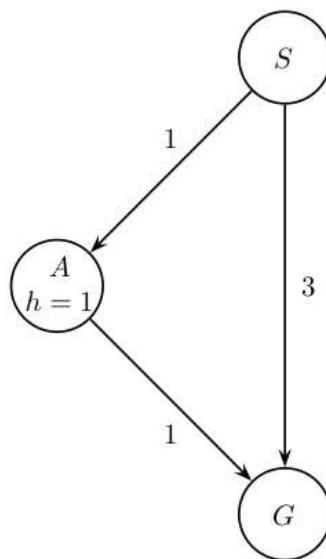
With the above implementation a node that reaches a goal state may sit on the fringe while the algorithm continues to search for a path that reaches a goal state. Let's consider altering the algorithm by testing whether a node reaches a goal state when inserting into the fringe. Concretely, we add the line of code highlighted below:

```

function EARLY-GOAL-CHECKING-GRAPH-SEARCH(problem, fringe, strategy) return a solution, or failure
  closed ← an empty set
  fringe ← INSERT(MAKE-NODE(INITIAL-STATE[problem]), fringe)
  loop do
    if fringe is empty then return failure
    node ← REMOVE-FRONT(fringe, strategy)
    if GOAL-TEST(problem, STATE[node]) then return node
    if STATE[node] is not in closed then
      add STATE[node] to closed
      for child-node in EXPAND(STATE[node], problem) do
        if GOAL-TEST(problem, STATE[child-node]) then return child-node
        fringe ← INSERT(child-node, fringe)
    end
  end

```

Now, we've produced a graph search algorithm that can find a solution faster. However, In doing so we might have affected some properties of the algorithm. To explore the possible differences, consider the example graph below.



If using EARLY-GOAL-CHECKING-GRAPH-SEARCH with a Uniform Cost node expansion strategy, which path, if any, will the algorithm return?

☒ S-G ✓

☐ S-A-G

☐ EARLY-GOAL-CHECKING-GRAPH-SEARCH will not find a solution path.

Submit

✓ Correct (2/2 points)

problem

2/2 points (ungraded)

If using EARLY-GOAL-CHECKING-GRAPH-SEARCH with an A* node expansion strategy, which path, if any, will the algorithm return?

☒ S-G ✓

☐ S-A-G

☐ EARLY-GOAL-CHECKING-GRAPH-SEARCH will not find a solution path.

Submit

✓ Correct (2/2 points)

problem

2/2 points (ungraded)

Assume you run EARLY-GOAL-CHECKING-GRAPH-SEARCH with the Uniform Cost node expansion strategy, select all statements that are true.

☒ The EXPAND function can be called at most once for each state.

☒ The algorithm is complete.

☐ The algorithm will return an optimal solution.

✓

Submit

✓ Correct (2/2 points)

problem

2/2 points (ungraded)

Assume you run EARLY-GOAL-CHECKING-GRAPH-SEARCH with the A* node expansion strategy and a consistent heuristic, select all statements that are true.

☒ The EXPAND function can be called at most once for each state.

☒ The algorithm is complete.

☐ The algorithm will return an optimal solution.



Submit

✓ Correct (2/2 points)
