Cole Feely and Zachary Zhao
Tue. Feb 8, 2022
CS 383
Professor Matthew Rattigan

Homework 1 Primer

1. What guarantees, if any, does Breadth-first search make, assuming a finite state space? Under what conditions do these guarantees hold?

Breadth-first search guarantees that the search will be complete and optimal, meaning that a solution will always be found and the solution found will be the best solution. However, the search will only be complete under the condition that the search space is finite and the search will only be optimal if the cost to move between each state is equal.

2. If every state has b successors, and the goal state is at depth d, what is the upper limit of the number of states Breadth-first search needs to add to its frontier?

In the worst case scenario of Breadth-first search, all the states at the depth of the goal state will need to be stored:

$$b^1 + b^2 + b^3 + b^4 + \ \dots \ + \ b^d$$

which will be $O(b^d)$ as for each level of depth, b states will be incorporated onto the frontier. This will be the upper limit on the number of states on the frontier : O(b^d).

3. Uniform-cost search is optimal, but Greedy best-first search is not. Given that both have exponential time complexity, why would you ever use Greedy?

Greedy best-fit search can yield much faster results, i.e. lower runtime, given that the problem being addressed has a good heuristic function. This would drastically reduce the number of node traversals necessary to reach the goal.

4. Which property or properties of A* is/are violated if the heuristic used is not admissible and consistent?

If the heuristic used is not admissible and consistent, the A* search will no longer be optimal.

5. If $h(s) = 0$ for all states s, is h admissible? What other search strategy that will behave the same as A* paired with s?

When $h(s) = 0$ for all states s, h is admissible because this means that $h(s)$ will be greater than or equal to the cost to get from any state, s, to the goal state meaning $h(s) \leq h^*(s)$. Uniform-cost search will behave the same as A* paired with s.

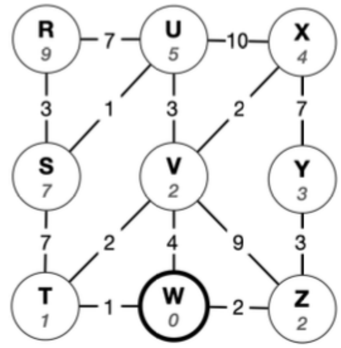
6. Given two admissible heuristics, h1(s) and h2(s), if h1(s) > h2(s) for all s, which heuristic will perform better? Why?

Consider the ideal function hi(s) that the two other admissible heuristics h1(s) and h2(s) are being compared to. Say for any finite value q, hi(q) = p. If we plug this value of p into our given heuristics: h1(q), h2(q), we know that h1(q) > h2(q) from the given information. Because h1(q) is greater than h2(q), we know that h1(q) will be closer to p than h2(q). Therefore h1(q) will perform better as h2(s) < h1(s) ≤ hi(s) for all s.

7. Explain why storing the frontier or explored states in a linked list is a bad idea for any best-first search (Uniform-cost, Greedy best-first, A*).

When searching for the next node during any best-first search, if a linked list is used, the entire linked list will need to be traversed in order to find the next best node. This would greatly decrease runtime efficiency when a priority queue would perform this function better. A priority queue would eliminate the need to traverse an entire list as it could just use the item at the top of the list to conduct the search.

8. The graph on the right represents a state space with states (nodes) **R-Z**. Possible transitions between states are represented by the edges in the graph, and numbers along the edges show the cost of each transition. The numbers inside the circles indicate the value of an admissible and consistent heuristic function that estimates the path cost to state W.



Consider a (graph-based) search with start state **R** and goal state **W.** For each of the following search strategies, indicate the order that the states will be popped from the frontier data structure and expanded, along with the solution path that will be returned. Show your work on the next page.

a. Breadth-first search

Frontier:	Explored:
R	
S, U	R
U, T	R, S
T, V, X	R, S, U
V, X, W	R, S, U, T

Goal Found

X, W, Z	R, S, U, T, V
W, Z, Y	R, S, U, T, V, X
Z, Y	R, S, U, T, V, X, W

Pop order: R, S, U, T

Solution: R →S → T → W

b. Uniform Cost Search

This search chooses the lowest cost child to explore

Frontier:	Explored:
R(0)	
S(3), U(7)	R
U(4), T(10)	R, S
V(7), T(10), X(14)	R, S, U
T(9), X(9), W(11), Z(16)	R, S, U, V
X(9), W(10), Z(16)	R, S, U, V, T
W(10), Z(16), Y(16)	R, S, U, V, T, X
Y(16)	R, S, U, V, T, X, W

Pop order: R, S, U, V, T, X, W

Solution: R $\rightarrow$ S$\rightarrow$ U $\rightarrow$ V $\rightarrow$ T $\rightarrow$W

c. Greedy best-fit search

Frontier:	Explored:
R(9)	
U(5), S(7)	R
V(2), X(4), S(7)	R, U
W(0), T(1), X(4), S(7)	R, U, V
<Doesn't matter>	R, U, V, W

Pop order: R, U, V, W

Solution: R→ U→ V → W

d. A* search

Frontier:	Explored:
R(9)	
S(10), U(12)	R
U(12), T(18)	R, S
V(17), T(18), X(26)	R, S, U
T(18),W(21), X(23), Z(28)	R, S, U, V
W(19), X(23), Z(28)	R, S, U, V, T
<Doesn’t matter>	R, S, U, V, T, W

Pop order: R, S, U, V, T, W

Solution: R → S → U→ V→ T→ W