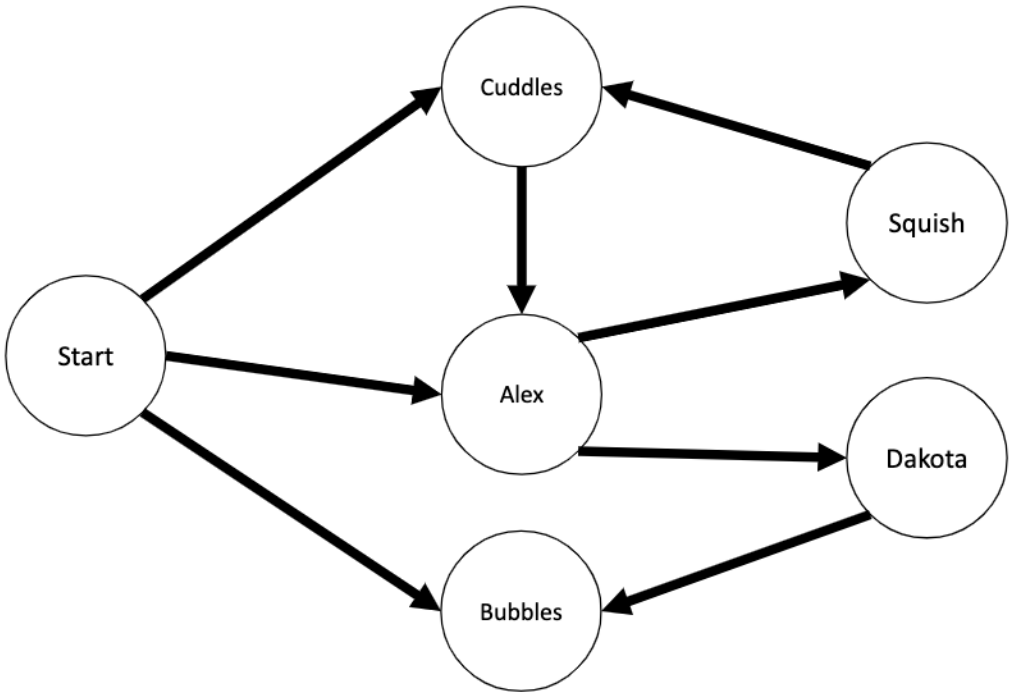
CS 188 Spring 2021 Introduction to Artificial Intelligence

Final Review Search

Q1. Search: Snail search for love

Scorpblorg the snail is looking for a mate. It can visit different potential mates based on a trail of ooze to nearby snails, and then test them for chemistry, as represented in the below graph, where each node represents a snail. In all cases, nodes with equal priority should be visited in alphabetical order.



(a) Simple search

In this part, assume that the only match for Scorpblorg is Squish (i.e. Squish is the goal state). Which of the following are true **when searching the above graph**?

(i) BFS Tree Search expands more nodes than DFS Tree Search ◯ True ◯ False

(ii) DFS Tree Search finds a path to the goal for this graph ◯ True ◯ False

(iii) DFS Graph Search finds the shortest path to the goal for this graph ◯ True ◯ False

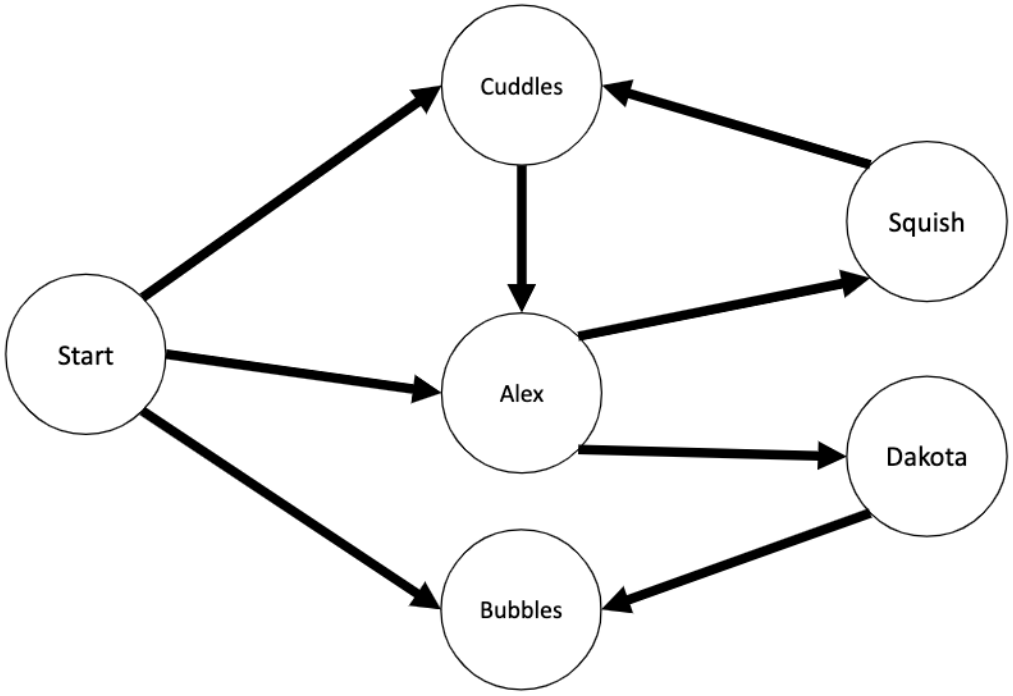
(iv) If we remove the connection from Cuddles → Alex, can DFS Graph Search find a path to the goal for the altered graph? ◯ Yes ◯ No

(b) Third Time's A Charm

Now we assume that Scorpblorg's mate preferences have changed. The new criteria she is looking for in a mate is that she has **visited the mate twice before** (i.e. when she visits any state for the third time, she has found a path to the goal).

(i) What should the most simple yet sufficient new state space representation include?

- ☐ The current location of Scorpblorg
- ☐ The total number of edges travelled so far
- ☐ An array of booleans indicating whether each snail has been visited so far
- ☐ An array of numbers indicating how many times each snail has been visited so far
- ☐ The number of distinct snails visited so far



(The graph is copied for your convenience)

(ii) DFS Tree Search finds a path to the goal for this graph ○ True ○ False

(iii) BFS Graph Search finds a path to the goal for this graph ○ True ○ False

(iv) If we remove the connection from Cuddles → Alex, can DFS Graph Search finds a path to the goal for the altered graph? ○ Yes ○ No

We continue as in part (b) where the goal is still to find a mate who is visited for the third time.

(c) Costs for visiting snails

Assume we are using Uniform cost search and we can now add costs to the actions in the graph.

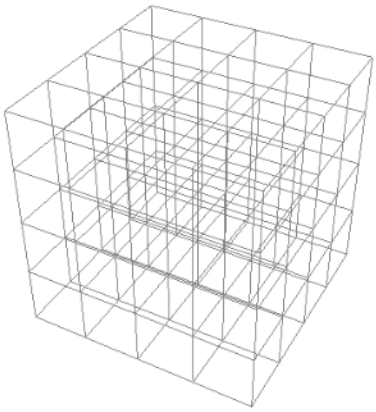
(i) Can one assign (non-negative) costs to the actions in the graph such that the goal state returned by UCS (Tree-search) changes? ○ Yes ○ No

(ii) Can one assign (potentially negative) costs to the actions in the graph such that UCS (Tree-search) will never find a goal state? ◯ Yes ◯ No

Q2. Search in 3D Maze

Imagine you are the Spider man. Your friend Superman was captured by the evil Spider man somewhere in this mystical 3D maze (Fig 1), (x_t, y_t, z_t). This maze is a *infinite* 3D grid world. You are located at $(0, 0, 0)$ right now and want to come up with a plan to rescue the Superman. Even though you are the Spider man, you can only travel along the wires, not through the space.

3D Maze



(a) What is the branch factor b in this space?

(b) How many distinct states can you reach at depth k?

(c) If you run BFS-tree search, how many nodes would you have expanded up to the goal state? What about BFS-graph search?

(d) Assume each edge has a cost of 1. Let the state be (u, v, w). Which of the following heuristics are admissible? Select all that apply.

- ☐ $h1(u, v, w) = \sqrt{(x_t - u)^2 + (y_t - v)^2 + (z_t - w)^2}$
- ☐ $h2(u, v, w) = |x_t - u| + |y_t - v| + |z_t - w|$
- ☐ $h3(u, v, w) = \sqrt{|x_t - u|} + \sqrt{|y_t - v|} + \sqrt{|z_t - w|}$
- ☐ $h1(u, v, w) = (x_t - u)^2 + (y_t - v)^2 + (z_t - w)^2$

(e) Approximately how many nodes would you expand if you use heuristic $h1$?

○ $|x_t y_t z_t|$ ○ $(x_t)^2 + (y_t)^2 + (z_t)^2$ ○ $\sqrt{(x_t - u)^2 + (y_t - v)^2 + (z_t - w)^2}$ ○ $|x_t| + |y_t| + |z_t|$

What about $h2$?

○ $|x_t y_t z_t|$ ○ $(x_t)^2 + (y_t)^2 + (z_t)^2$ ○ $\sqrt{(x_t - u)^2 + (y_t - v)^2 + (z_t - w)^2}$ ○ $|x_t| + |y_t| + |z_t|$

(f) If the evil Spider man destroys half of the links in this grid, would the heuristics h_1 and h_2 be admissible?

(g) In expectation (assume random tie-breaking), how many nodes would you expand before hitting the goal if you use UCS tree search, DFS graph search, or greedy search instead of A* search? Assume each path has cost 1 and heuristic $h1$.