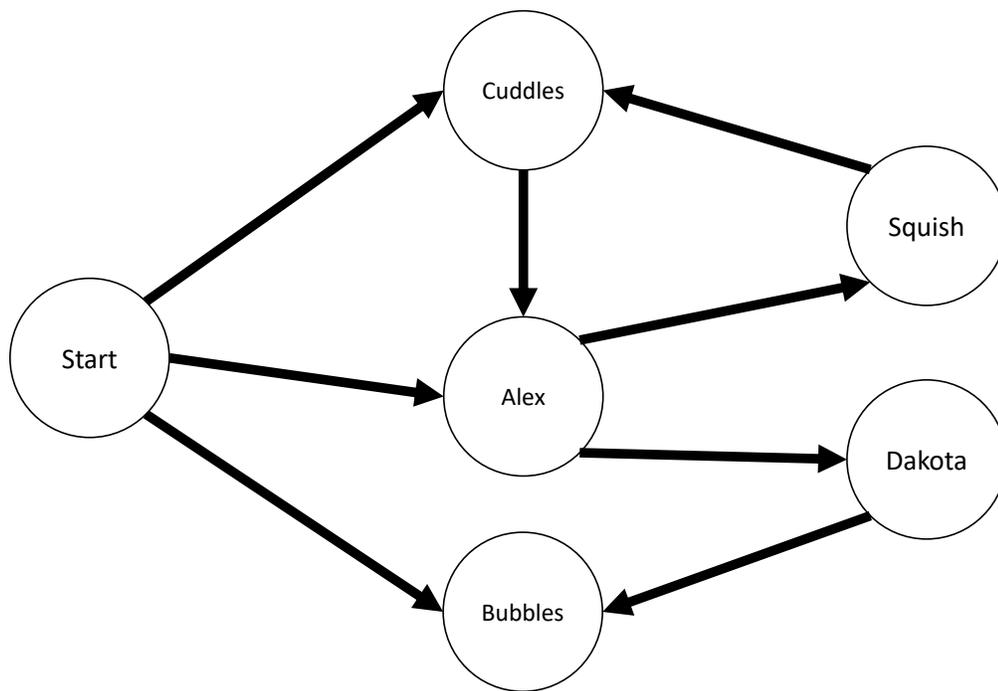


Q1. Search: Snail search for love

Scorpblog 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**

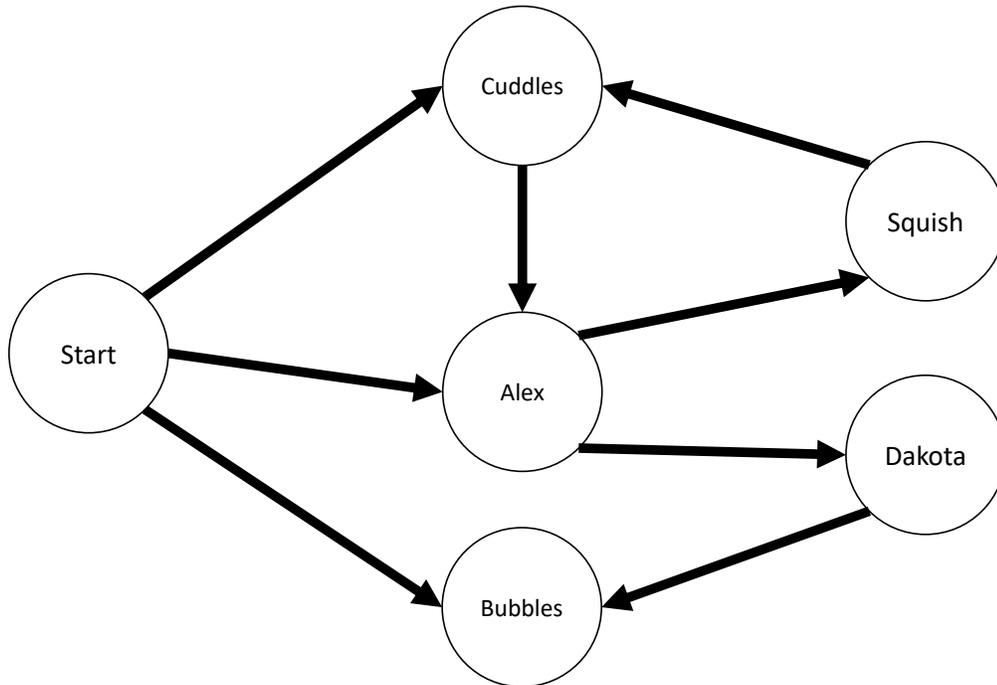
In this part, assume that the only match for Scorpblog is Squish (i.e. Squish is the goal state). Which of the following are true **when searching the above graph**?

- (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. Power Pellets

Consider a Pacman game where Pacman can eat 3 types of pellets:

- Normal pellets (n-pellets), which are worth one point.
- Decaying pellets (d-pellets), which are worth $\max(0, 5 - t)$ points, where t is time.
- Growing pellets (g-pellets), which are worth t points, where t is time.

The pellet's point value stops changing once eaten. For example, if Pacman eats one g-pellet at $t = 1$ and one d-pellet at $t = 2$, Pacman will have won $1 + 3 = 4$ points.

Pacman needs to find a path to win at least 10 points but he wants to minimize distance travelled. The cost between states is equal to distance travelled.

(a) Which of the following must be including for a minimum, sufficient state space?

- Pacman's location
- Location and type of each pellet
- How far Pacman has travelled
- Current time
- How many pellets Pacman has eaten and the point value of each eaten pellet
- Total points Pacman has won
- Which pellets Pacman has eaten

(b) Which of the following are admissible heuristics? Let x be the number of points won so far.

- Distance to closest pellet, except if in the goal state, in which case the heuristic value is 0.
- Distance needed to win $10 - x$ points, determining the value of all pellets as if they were n-pellets.
- Distance needed to win $10 - x$ points, determining the value of all pellets as if they were g-pellets (i.e. all pellet values will be t .)
- Distance needed to win $10 - x$ points, determining the value of all pellets as if they were d-pellets (i.e. all pellet values will be $\max(0, 5 - t)$).
- Distance needed to win $10 - x$ points assuming all pellets maintain current point value (g-pellets stop increasing in value and d-pellets stop decreasing in value)
- None of the above

(c) Instead of finding a path which minimizes distance, Pacman would like to find a path which minimizes the following:

$$C_{new} = a * t + b * d$$

where t is the amount of time elapsed, d is the distance travelled, and a and b are non-negative constants such that $a + b = 1$. Pacman knows an admissible heuristic when he is trying to minimize time (i.e. when $a = 1, b = 0$), h_t , and when he is trying to minimize distance, h_d (i.e. when $a = 0, b = 1$).

Which of the following heuristics is guaranteed to be admissible when minimizing C_{new} ?

- $mean(h_t, h_d)$
- $min(h_t, h_d)$
- $max(h_t, h_d)$
- $a * h_t + b * h_d$
- None of the above