**AI Exam 2020 Preparation**

**Search**

There are 2 types of search algorithms:

1. Uninformed search (Blind search).
2. Informed search (Heuristic search).

Uninformed search:

* Breadth first search
* Uniform cost search
* Depth first search
* Depth limited search
* Iterative deeping depth first search
* Bidirectional search

Uninformed search does not contain any domain knowledge such as closeness or the location of the goal.

It operates in a brute force way, as it only includes info about how to traverse the tree and how to identify leaf and goal nodes.

Uninformed search applies a way in which the search tree is searched without any info about the search space like initial state operator and test for the goal so that is why it is called blind search. It examines each node until it achieves its goal node.

**Breadth First Search (BFS)**

* Searches breadthwise in a tree or graph.
* Starts searching from root node of the tree and searches all successor nodes at the current level before moving on to the next level.
* Implemented using FIFO queue.
* Example of a general graph search algorithm.
* Advantages
  + Provides a solution if any solution exists.
  + If there are more than one solution for a given problem, the BFS will provide minimal solution which requires the least number of steps.
* Disadvantages:
  + Requires a lot of memory since each level of the tree must be saved into mem to search the next level.
  + BFS needs lots of time if the goal is far away from the root node.

**Depth First Search (DFS)**

* Recursive alg for traversing a tree or graph
* Starts from the root and follows each path to its greatest depth before moving to the next

Informed search:

* Best first search
* A\* search
* Ao\* search
* Problem reduction
* Hill climbing

Informed search uses domain knowledge, the problem info is available which can guide the search.

Informed search strategies can find a solution more efficiently than an uninformed.

Aka heuristic search.

Heuristic does not always guarantee the best solution but guaranteed to find the solution in reasonable time.

It can solve more complex problems which could not be solved in another way.

**2018 Q1**

**A picture containing bird

Description automatically generated**

A picture containing game

Description automatically generated

A picture containing knife, bird

Description automatically generated

A picture containing bird

Description automatically generatedA close up of a piece of paper

Description automatically generated

Should be modified so that the head of the frontier has the lowest cost q-value..

f(x) = cost(x) + h(x)

[Head|Tail]… f(Head) <= f(T) for all values of T ∊ Tail

NO ANSWER FOR IMPLEMENTATION

A close up of a logo

Description automatically generated

Basically means that if there is a solution, A\* will always find it, and it will have the minimal cost value. A heuristic function is said to be admissible if it never overestimates the cost of reaching the goal, i.e. the cost it estimates to reach the goal is not higher than the lowest possible cost from the current point in the path.

A picture containing knife, table

Description automatically generated

1. Branching factor is finite.

2. Arc costs are bounded above zero.

3. h(n) is an underestimate of the length of the shortest path from n to a goal node.

Yes these conditions do guarantee that A\* will terminate, since the arc costs are bounded above zero.

A picture containing knife, table

Description automatically generated

A picture containing bird

Description automatically generatedA screenshot of a cell phone

Description automatically generatedBFS is admissible since it is guaranteed to find optimal solutions provided that any exist.

No, A\* is not admissible under this setup, since branching factor is infinite. Maybe also because h(n) is not an underestimate of the length of the shortest path from n to a goal node?

A picture containing bird, tree

Description automatically generated

A\* still won’t be admissible under this setup since the branching factor is still infinite.

**2018 Q2**

**A picture containing knife

Description automatically generated**

Policy is a mapping s->a, that tells us what action to take for all states.

3 states, 2 actions

3\*2 = 6 policies

A close up of text on a white surface

Description automatically generated

y-optimal policy is the best policy, the best action for us to take for all states.