State space approach:

Defining State & Search Space

A state is a representation of problem elements at a given moment.

A State space (Search space) is the set of all states reachable from the initial state, it can be represented by an oriented graph which the nodes are states and the arcs are actions.

Ref: <https://medium.com/@dpthegrey/defining-state-search-space-c8b5ae3c56a>

Graphs:

A graph is a nonlinear data structure that contains nodes and arcs; a node is just a single unique value while an arc represents a connection between these nodes.

Graphs can be weighted which means the arc has some additional data about the relationship like the distance between two locations.

Graph representation:

There are several ways of representing a graph as examples: with a two dimensional array, adjacency matrix or adjacency list and many more.

Graph traversal:

Graph traversal is a technique used for searching a node in a graph, there are two graph traversal techniques and they are as follows:

DFS (Depth First Search).

BFS (Breadth First Search).

DFS (Depth First Search):

In DFS graph traversal we start with any random node , go to its first child then its first child and we keep doing so as long as we can, when there are no more children backtrack to the last node and continue the process, this algorithm is usually implemented with a recursive function using stack data structure. DFS returns at the end a graph without loops.

Depth first search algorithm:

Define a stack with maximum size of total number of nodes.

Select any node as a starting point, visit that node and push it to the stack.

Visit first child of the node which is at the top of the stack and push it to the stack.

Repeat step 3 until there are no more new children to be visited from the node on the top of the stack.

Pop one node from the stack

Repeat step 3, 4 and 5 until the stack becomes empty.

BFS (Breadth First Search):

In BFS graph traversal we start with any random node, we add all its direct children to a queue, once they’ve been visited we move to their children (children’s children) and we continue following this pattern, at the end BFS algorithm returns a graph with no loops.

Breadth first search algorithm:

Define a queue with maximum size of total number of nodes.

Select any node as a starting point, visit that node and insert it into the queue.

Visit all the children of the node which is at front of the queue and insert them into the queue.

When there are no new children to be visited from the node at front of the queue then delete it.

Repeat steps 3 and 4 until queue becomes empty.