

# Artificial Intelligence

## Assignment 01

### Names:

- 1) Ahmed Mohamed EL-Bawab (08)
- 2) Khalil Ismail Khalil (23)

### 1) DFS:

#### 1- Data Structure:

we used two data structure {Stack, Set}, using Stack to store expanded nodes and Set to store visited nodes.

#### 2- Algorithm:

- 1- start with pushing initial state in Stack.
- 2- check if Stack is not empty.
- 3- pop state from Stack.

- 4- add this state to set.
- 5- check if state equal goal state.
- 6- if not equal expand this state and get their children.
- 7- check for each child if not exist in Stack and Set, push this child in Stack.
- 8- repeat from step 2 to step 7.

## 2) BFS:

### 1- Data Structure:

we used two data structure {Queue, Set}, using Queue to store Expanded Nodes and Set to store Visited Nodes.

### 2- Algorithm:

- 1- start with enqueue initial state in Queue.
- 2- check if Queue is not empty.
- 3- dequeue state from Queue
- 4- add this state to set.
- 5- check if state equal goal state.
- 6- if not equal expand this state and get their children.
- 7- check for each child if not exist in Queue and Set, enqueue this child in Queue.
- 8- repeat from step 2 to step 7.

### 3) A\* Search:

#### 1- Data Structure:

we used two data structure {Heap, Set}, using Heap to store Expanded Nodes and Set to store Visited Nodes.

#### 2- Algorithm:

- 1- calculate cost of initial state.
- 2- insert initial state with its cost in Heap.
- 3- check if Heap is not empty.
- 4- get state with min cost from Heap
- 5- add this state to set.
- 6- check if state equal goal state.
- 7- if not equal expand this state and get their children.
- 8- check for each child if not exist in Set, calculate cost of this child.
- 9- check if child is not exist in Heap, insert it in the Heap.
- 10- if the child exist in the Heap, decrease its cost as possible.
- 11- repeat from step 2 to step 10.