ASSIGNMENT - 1

AIM : Implementation of uninformed search methods with eight puzzle problem.

OBJECTIVE :

- To understand eight puzzle Problem.
- To implement uninformed search method with eight puzzle problem.

SOFTWARE REQUIREMENTS :

- Linux Operating System
- Java Compiler
- Eclipse IDE

MATHEMATICAL MODEL :

Consider a following set theory notations related to a program. The mathematical model M for Eight puzzle problem is given as below,

M={S,So,A,G}

Where,

S=State space.A state description specifies the location of each of the eight Ides and the blank in one of the nine squares.

So= Initial State.Any state can be designated as the initial state. Any given goal can he reached Front exactly half of the possible initial states.

A=Set of Actions/Operators.The simplest formulation defines the actions as movements of the blank space Left, Right, Up, or Down. Different subsets of these are possible depending on where the blank is.

G=Goal state., In case of Eight puzzle user can decide a goal state.

THEORY :

UNINFORMED SEARCH :

An uninformed search is the strategy that do not have additional information about states beyond that provided in the problem definition. All they can do is generate successors and distinguish a goal state from a non-goal state. Uninformed search is also known as 'Blind search'.

UNINFORMED SEARCH STRATERGIES:

1. Breadth First Search :
   In BFS root node is expanded first ,then all the successors of root node are expanded and then their successor and so on.That is the node are expanded levelwise starting at root level.

2. Depth First Search:
   DFS always expands the deepest node in the current explored node set of the search tree.The search goes in to depth until their is no more successor nodes.As this node are expanded ,they are dropped from the set ,so then the search "back up" to the next sallowest node that still has unexplored successors.

3. Depth Limited Search:
   If we can limit the depth of search tree to a certain level then searching will be more efficient.This removes the problem of unbounded tress.Such a kind of DFS where in the depth is limited at a certain level is called as depth limited search(DLS).It solves infinite path problem.

4. Bidirectional Search:
   As the name suggest bidirectional that is two directional searches are made in this searching techniques. One is the forward search which starts from initial state and the other is the backward search which starts from goal state.The two searches stop when both the searches meet in the middle.

5. Uniform Cost Search :
   Uniform Cost Search expands the node 'n' with the lowest path cost. Uniform Cost Search does not care about the number of steps a path has, instead it considers their total cost.Therfore,their is a chance of getting stuck in an infinite loop if it ever exands a node. That has a zero-cost action leading back to the same states.

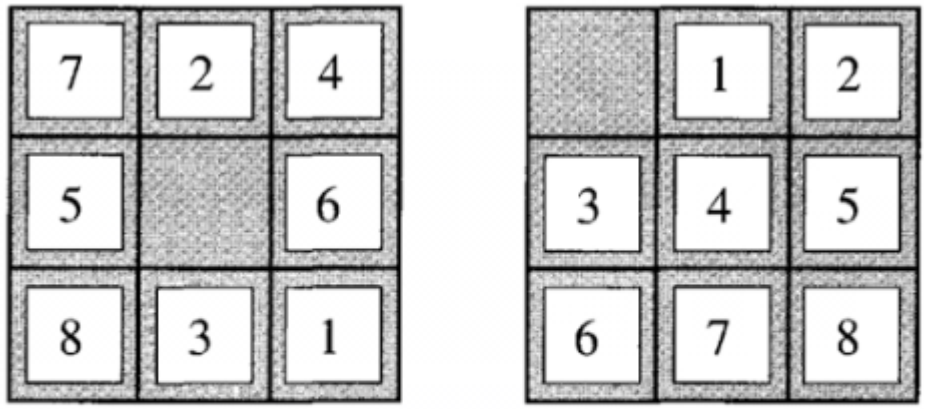


Fig : A typical instance of 8-Puzzle problem

8-Puzzle Problem :

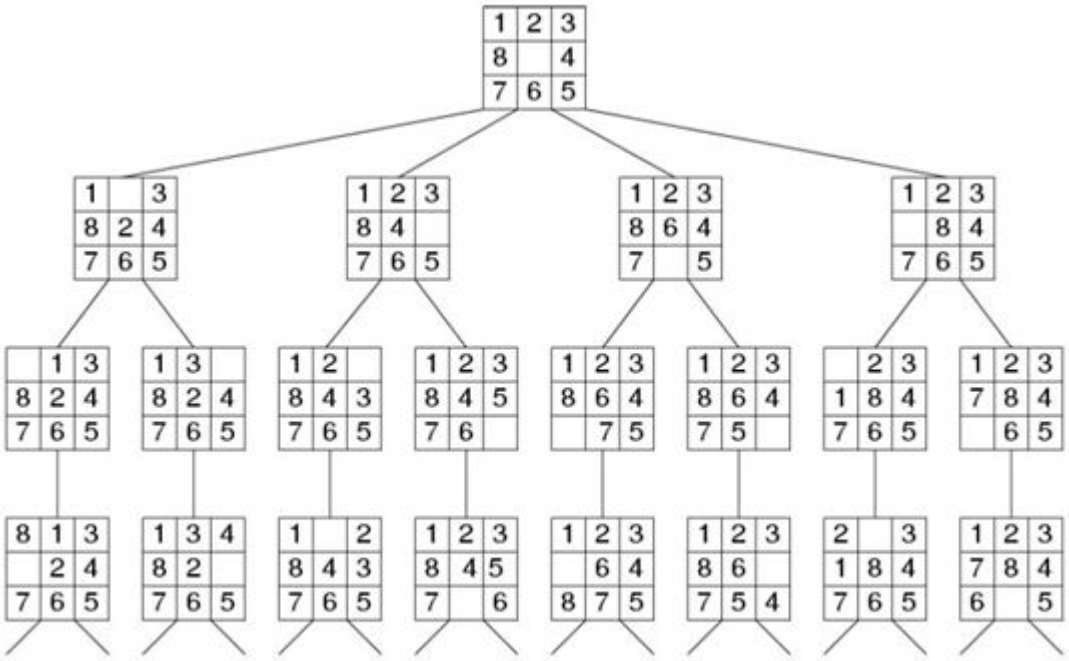
The 8-puzzle is an instance which consists of a 3x3 board with eight numbered tiles and a blank space. A tile adjacent to the blank space can slide into the space. The object is to reach a specified goal state, such as the one shown in the above figure.

The 8-puzzle belongs to the family of sliding-block puzzles, which are often used as test problems for new search algorithms in AI. This family is known to be NP-complete, so one does not expect to find methods significantly better in the worst case than the search algorithms described in this chapter and the next. The 8-puzzle has 91/2 =181, 440 reachable states and is easily solved. The 15-puzzle (on a 4 x 4 board) has around 1.3 trillion states, and random instances can be solved optimally in a few milliseconds by the best search algorithms. The 24-puzzle (on a 5 x 5 board) has around 1025 states, and random instances take several hours to solve optimally.

The problem statement can be covered into an standard formulation and specified as follows:

1. **States**
   There are eight tiles and one vacant position.State specifies the position of all the eight position of one vacant tile.

2. **Initial State**



In 8 puzzle problem, you can consider any state as initial state from which you start the game.

3. **Successor function**
   Successor function generates the next state from existing state from an existing state by taking the specified set of action .The permissible and possible set of action .

4. **Goal State**
   This is a function that verifies that weather the current state is goal state or not (as in figure).

5. **Path Cost**
   Each step can be assigned a numerical values called cost.It we assign each step a cost 1, then path cost will be the the total number steps from initial state goal state.

CONCLUSION :

Thus, we have implemented an uninformed search method for 8 puzzle problem.

Roll No.	Name of Student	Date of Performance	Date of Submission	Sign.
		/ /2015	/ /2015	