South China University of Technology

《Course》Experiment Report

Experiment Title： Experiment 4 Tree Search Algorithm

Name： Yeo King Student ID： 201969990051

Class： CST 2019 International Group： -

Collaborator： -

Teacher： Mr Yan Xiaoyang

|  |
| --- |
| **Description** |
| 【Objective and Requirement】  Objective：  （1）Understand the Tree Search algorithm.  （2）Learn the solution expression of some problems by A Tree data structure.  （3）Understand what kinds of problems can be solved by this algorithm.  （4）Realize the source code for this algorithm.  （5）Understand different search strategies, such as: the Breadth-first search**(**BFS) ,the Depth-first search**(**DFS) and the Best-first search**(**BFS) Search strategies etc., and compare the difference of results solved by them.    Requirement：   1. The template should be used for all kinds of data type, such as: integer, real, double, etc. in the program; 2. Programs should be made by Object-Oriented Programming (OOP) method; 3. Use different strategies to perform this algorithm, such as: the Breadth-first search(BFS) ,the Depth-first search(DFS) and the Best-first search(BFS) Search strategies etc.. 4. And compare the result difference solved by them. 5. Write down the report in which there should be the execution results of the program.   【Environment】  Operating System：Windows |
| **Content** |
| To solve 8 puzzle with tree search I have used Uniform Cost Search compare with A\* with Manhattan distance heuristic  Initial Goal  Uniform Cost Search is uninformed search, it doesn't use any domain knowledge. It expands the least cost node, and it does so in every direction because no information about the goal is provided. It can be viewed as a function **f(n) = g(n)** where g(n) is a path cost ("path cost" itself is a function that assigns a numeric cost to a path with respect to performance measure, e.g. distance in kilometers, or number of moves etc.). It simply is a cost to reach node n.  While for A\* Algorithm is a are informed search algorithms algorithm that relies on an open list and a closed list to find a path that is both optimal and complete towards the goal. It works by combining the benefits of the uniform-cost search and greedy search algorithms. A\* makes use of both elements by including two separate path finding functions in its algorithm that take into account the cost from the root node to the current node and estimates the path cost from the current node to the goal node. The first function is g(n), which calculates the path cost between the start node and the current node. The second function is h(n), which is a heuristic to calculate the estimated path cost from the current node to the goal node. **F(n) = g(n) + h(n)**. It represents the path cost of the most efficient estimated path towards the goal. A\* continues to re-evaluate both g(n) and h(n) throughout the search for all of the nodes that it encounters in order to arrive at the minimal cost path to the goal. This algorithm is extremely popular for pathfinding in strategy computer games.  The heuristic should be admissible otherwise A\* will find a suboptimal solution. The queue should not be ordinated by the distance of the next node but using the heuristic that will put as next node the most promising one. The next node may be the most distant from the current one but at the same time it can be the nearest one to the destination. In this case A\* using Manhattan distance heuristic The Manhattan distance of a puzzle is defined as:  h ( n ) = ∑ all tiles d i s t a n c e ( tile, correct position ) {\displaystyle h(n)=\sum \_{\text{all tiles}}{\mathit {distance}}({\text{tile, correct position}})}  Consider the puzzle in which the player wishes to move each tile such that the numbers are ordered. The Manhattan distance is an admissible heuristic in this case because every tile will have to be moved at least the number of spots in between itself and its correct position  Below are the results of program execution |
| **Conclusion** |
| I have learnt to understand different tree search algorithm to solve 8 puzzle problem .By using Uniform Cost search and A\* algorithm to solve the said problem. |
| **Teacher’s Comments and Score** |
| Comment：  Score：           Signature：                                                 Date： |