**VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY**

**INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE & ENGINEERING**

Logo

Description automatically generated

**INTRODUCTION TO ARTIFICIAL INTELLIGENCE**

Course by Dr. Nguyen Trung Ky

**LAB #2:**

**UNINFORMED SEARCH IN PAC-MAN**

Student's Name: Dang Nhat Huy

Student's ID: ITITIU20043

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Depth-First Search | | | Breadth-First Search | | | Uniform-Cost Search | | |
| Maze | #nodes explored | Solution length | Is it optimal? | #nodes explored | Solution length | Is it optimal? | #nodes explored | Solution length | Is it optimal? |
| tiny | 15 | 10 | No | 16 | 8 | Yes | 16 | 8 | Yes |
| medium | 146 | 130 | No | 275 | 68 | Yes | 275 | 68 | Yes |
| big | 390 | 210 | No | 620 | 210 | Yes | 620 | 210 | Yes |

Starting at the root node, the DFS search algorithm follows each branch as far as it can go before looping back. To maintain track of the nodes that must be visited, it makes use of a stack data structure. Although DFS is user-friendly and useful for solving problems requiring a network scan, it may not always identify the optimum solution and may become stuck in an unending loop.

Before moving on to the next node, the BFS checks each one at the current depth, starting at the root node. It keeps track of the places that need to be visited by using a queue data structure. BFS guarantees to find the fastest path between two nodes in an unweighted network and is useful when the solution is close to the root node.

The UCS search method finds the best way by lowering the total cost of the route while taking into account the cost of each edge in the network. It keeps track of the websites that need to be accessed using a priority list data structure. Even while UCS will always discover the optimal solution, it could take longer than other approaches, especially in large networks with many of links.

The time complexity and volume complexity of DFS and BFS are O(V + E), where V is the number of nodes and E is the number of edges in the network. In contrast, UCS has an O((V + E) log V temporal complexity and employs a priority list. In terms of space complexity, DFS, BFS, and UCS require O(V) space, whereas O(V + E) space is needed by UCS.

To sum up, DFS, BFS, and UCS are all practical search algorithms, each with particular advantages and disadvantages. DFS is simple to use, but it might not always produce the optimal outcome. BFS guarantees the fastest path in non-weighted graphs, but in large networks, it could take longer. Although UCS may take longer than other techniques in large maps, it yields the greatest results. The appropriate method depends on the specific problem being solved as well as the characteristics of the network.