|  |  |  |  |
| --- | --- | --- | --- |
|  | Logo  Description automatically generated | | **Uninformed Search in Pac-Man.** |
| Lab #2 |
|  | **AI laboratory** |  | **Nguyen Thi Mai Huong – ITITIU19128**  **There are two exercises in this lab:**  **1. Depth-First Search (DFS)**  **2. Breadth-First Search (BFS)**  **3. Uniform-Cost Search (UCS)**  Based on the 3 search algorithms that I have implemented in lab 2, I triggered the below comparative obtained results table from these experiments:   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | Depth-First Search | | | Breadth-First Search | | | Uniform-Cost Search | | | | Maze | #nodes  explored | Sol  length | Is it  optimal? | #nodes  explored | Sol  length | Is it  optimal? | #nodes  explored | Sol  length | Is it  optimal? | | tiny | 15 | 10 | No | 15 | 8 | Yes | 15 | 8 | Yes | | medium | 146 | 130 | No | 269 | 68 | Yes | 269 | 68 | Yes | | big | 390 | 210 | Yes | 620 | 210 | No | 620 | 210 | No |  * Obviously from the three search strategies’ observation, we can conclude that the Uniform Cost Search brings the best time complexity in the case of different path costs in the maze, otherwise, it is similar to the breadth-first search. When all the path costs offer the same, BFS gives the best time complexity of O(b^d) but an exponential space complexity. In terms of space complexity, Depth-First Search gives a linear one, in the meanwhile, its time complexity is similar to the breadth-first search * In certain conditions, DFS is neither complete (e.g if the state space graph has cycles) nor optimal whereas the other two are complete and optimal * I compare these 3 different search algorithms and compare them in terms of performance, completeness, and optimality as shown in the below table:  |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | | Depth-First Search | Breadth-First Search | Uniform-Cost Search | | performance | Time complexity | O(**b**m) | O(**b**d) | O(**b**(1+C/ε)) | | Space complexity | O(bm) | O(**b**d) | O(**b**(1+C/ε)) | | completeness |  | Yes, if the tree is finite, otherwise no | Yes | Yes, if there is no negative cost | | optimality |  | No | Yes, if all the costs are the same | Yes |   In which,  + C is the cost of the optimal solution.  + ε is the least cost of actions.  + b is the branching factor of the search tree.  + m is the maximum depth of the tree.  + d is the depth at which the shallowest goal node is situated. |
|  |  |  |  |