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

**INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**Ảnh có chứa vòng tròn, biểu tượng, Nhãn hiệu, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.**

**ARTIFICIAL INTELLIGENCE**

**IT159IU**

**REPORT LAB 2**

**Instructor:**

**Dr. Nguyen Trung Ky**

**Dr. Ly Tu Nga**

**Nguyen Huynh Ngan Anh - ITDSIU23003**

1. **Setup:**

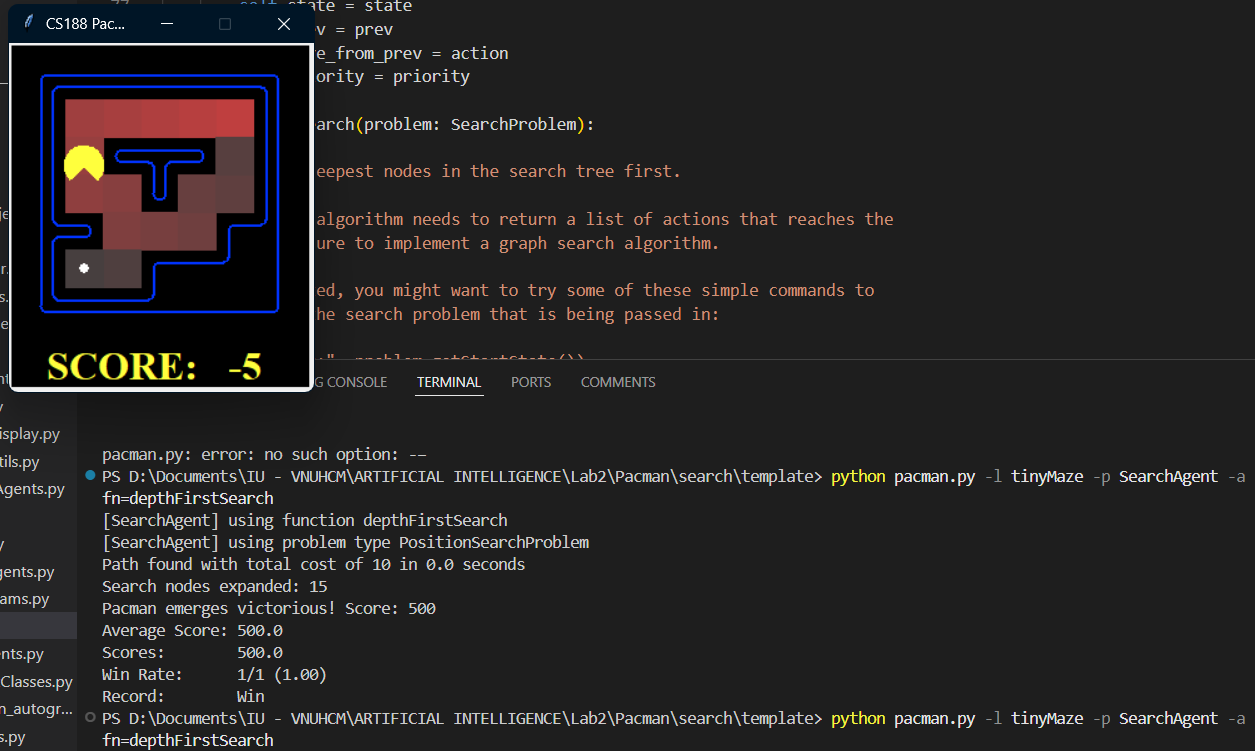
Tried running the below command to run the game

**Ảnh có chứa văn bản, Phông chữ, phần mềm, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.**

1. **Exercise 1:**

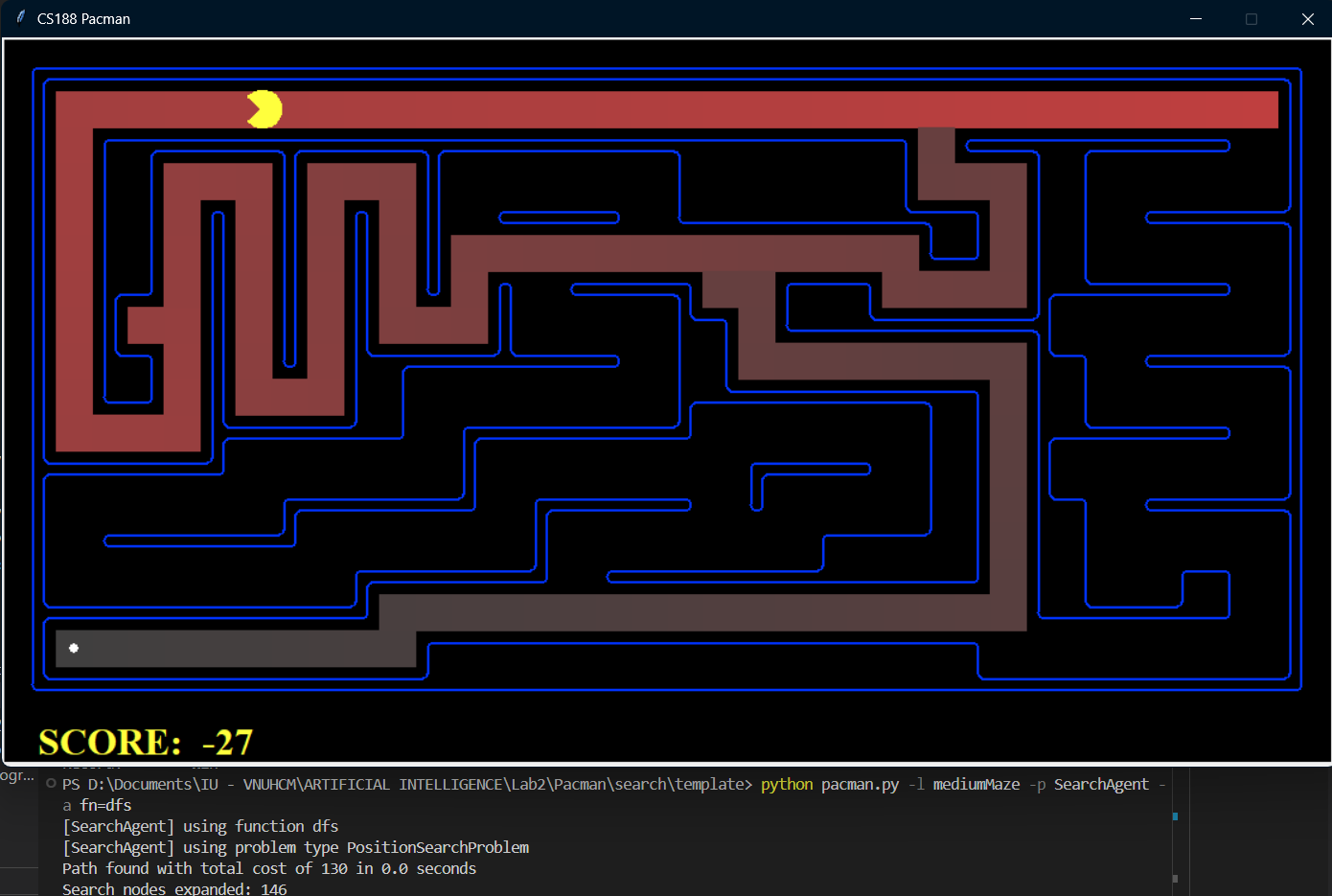
Running dfs algorithm for tinyMaze



Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Running dfs algorithm for mediumMaze



Ảnh có chứa ảnh chụp màn hình, văn bản

Nội dung do AI tạo ra có thể không chính xác.

Running dfs algorithm for bigMaze

Ảnh có chứa văn bản, ảnh chụp màn hình, mẫu

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa ảnh chụp màn hình, văn bản, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

1. **Exercise 2:**

Running bfs algorithm for tinyMaze, mediumMaze and bigMaze

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phần mềm đa phương tiện, thiết kế

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, Hệ điều hành

Nội dung do AI tạo ra có thể không chính xác.

1. **Exercise 3:**

Running ucs algorithm for tinyMaze, mediumMaze and bigMaze

Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Nội dung do AI tạo ra có thể không chính xác.

1. **Conclusion**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **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 | 15 | | 8 | Yes | 15 | 8 | Yes |
| **medium** | 146 | 130 | No | 269 | | 68 | Yes | 269 | 68 | Yes |
| **big** | 390 | 210 | No | 620 | | 210 | Yes | 620 | 210 | Yes |

The DFS search algorithm begins at the root node and proceeds as far as it can go along each branch before turning around. It employs a stack data structure to keep track of the nodes that need to be visited. DFS is easy to use and can be used to resolve issues that call for a scan of the complete network, but it might not always find the best answer and might become trapped in an endless cycle.

The BFS begins at the root node and examines every node at the current depth before going on to the following. It employs a queue data structure to maintain an account of the locations that need to be viewed. BFS is helpful when the answer is near the root node and is assured to identify the quickest route between two nodes in an unweighted network.

By reducing the route's overall cost while considering the cost of each edge in the network, the UCS search method determines the optimal path. It uses a priority list data structure to maintain an account of the sites that need to be viewed. Although UCS will always find the best answer, it may take longer than other methods, particularly in extensive networks with many links.

DFS and BFS have an O(V + E) time complexity and an O(V + E) volume complexity, in which V denotes the number of nodes and E represents the number of edges in the network. On the other hand, UCS uses a priority list and has a temporal complexity of O((V + E) log V). DFS and BFS require O(V) space, while UCS needs O(V Plus E) space in terms of space complexity.

In conclusion, DFS, BFS, and UCS are all helpful search algorithms, each with unique benefits and drawbacks. Although DFS is easy to use, it may not always yield the best result. BFS ensures the quickest route in unweighted graphs, but it may take longer in extensive networks. In big charts, UCS may be slower than other methods but provides the best outcome. The particular issue being addressed and the properties of the network determine which method should be used.

I try to visualize the searching algorithm to under the form of tree node to better understand how algorithms is optimal to each other

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Nội dung do AI tạo ra có thể không chính xác.

And modify the searching algorithm to return the required value for visualization

Ảnh có chứa văn bản, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

I test with dfs but the visualize seems not working right, I’ll try to modify it later

Ảnh có chứa biểu đồ, hàng, bản đồ

Nội dung do AI tạo ra có thể không chính xác.