CS 430: ARTIFICIAL INTELLIGENCE

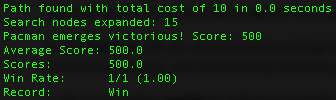
PROJECT 1 – PACMAN

Prof. Grissom

September 29, 2018

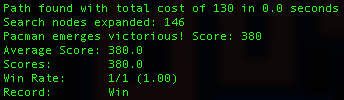
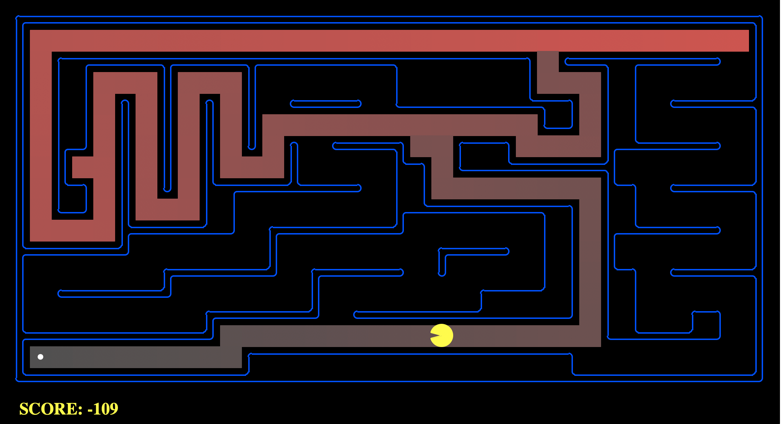
Gregory Montilla

1. Depth First Search
   1. tinyMaze



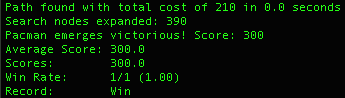
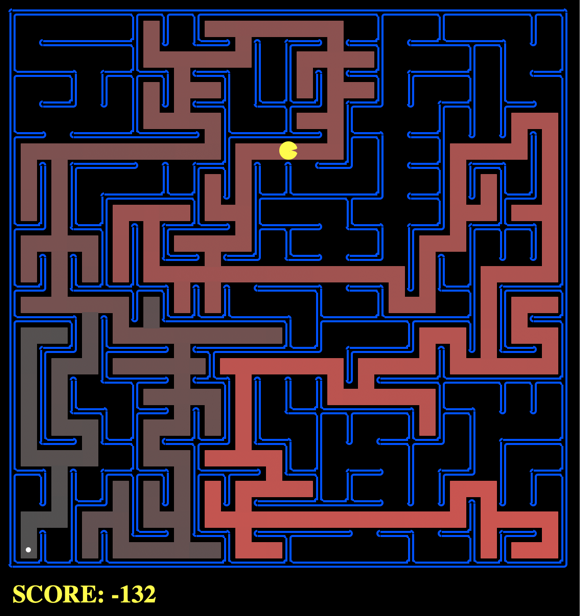
|  |  |
| --- | --- |
| Cost | 10 |
| Execution Time | 0.0 |
| Nodes Expanded | 15 |
| Score | 500 |

* 1. mediumMaze



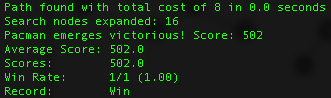
|  |  |
| --- | --- |
| Cost | 130 |
| Execution Time | 0.0 |
| Nodes Expanded | 146 |
| Score | 380 |

* 1. bigMaze



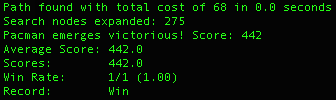
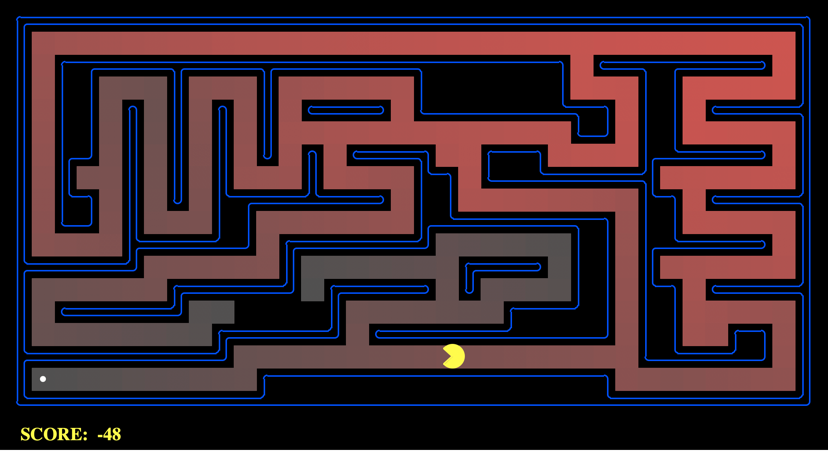
|  |  |
| --- | --- |
| Cost | 210 |
| Execution Time | 0.0 |
| Nodes Expanded | 390 |
| Score | 300 |

1. Breadth First Search
   1. tinyMaze



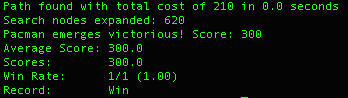
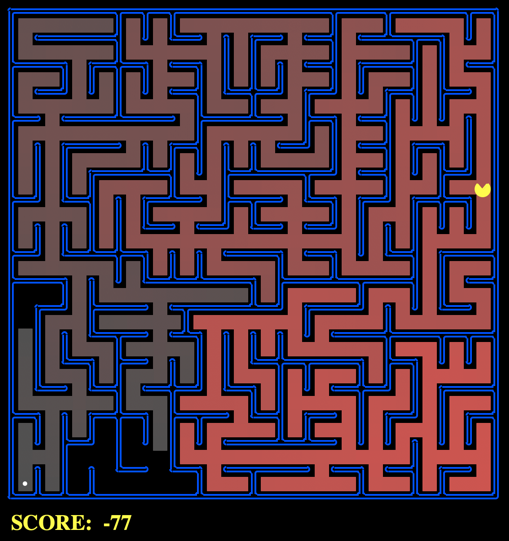
|  |  |
| --- | --- |
| Cost | 8 |
| Execution Time | 0.0 |
| Nodes Expanded | 16 |
| Score | 502 |

* 1. mediumMaze



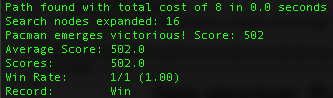
|  |  |
| --- | --- |
| Cost | 68 |
| Execution Time | 0.0 |
| Nodes Expanded | 275 |
| Score | 442 |

* 1. bigMaze



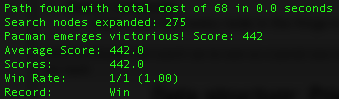
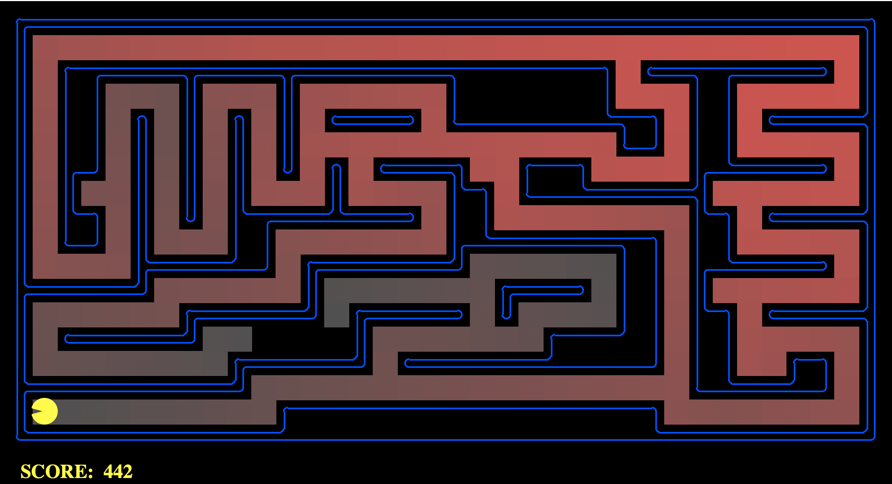
|  |  |
| --- | --- |
| Cost | 210 |
| Execution Time | 0.0 |
| Nodes Expanded | 620 |
| Score | 300 |

1. Uniform Cost Search
   1. tinyMaze



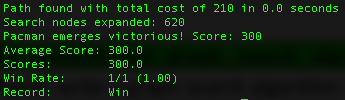
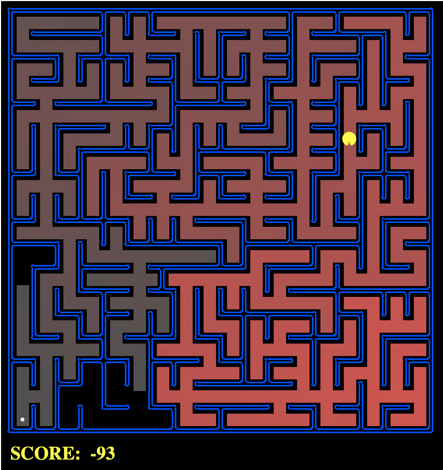
|  |  |
| --- | --- |
| Cost | 8 |
| Execution Time | 0.0 |
| Nodes Expanded | 16 |
| Score | 502 |

* 1. mediumMaze



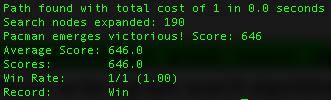
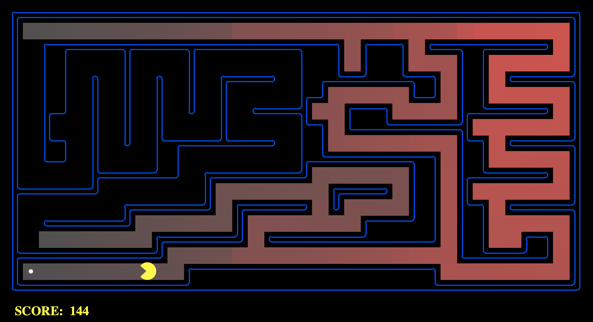
|  |  |
| --- | --- |
| Cost | 68 |
| Execution Time | 0.0 |
| Nodes Expanded | 275 |
| Score | 442 |

* 1. bigMaze



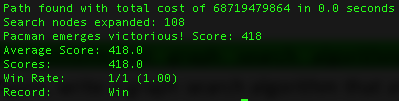
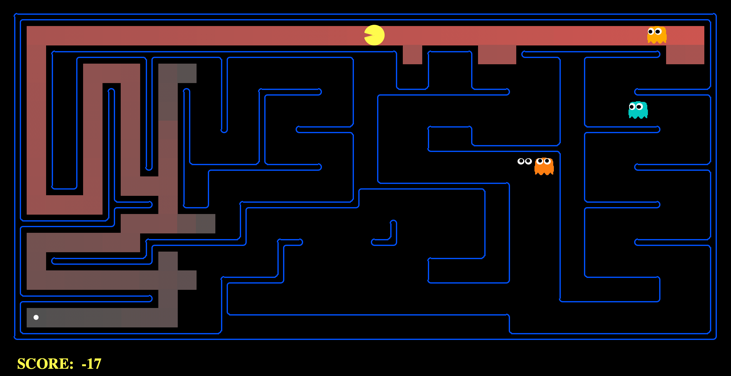
|  |  |
| --- | --- |
| Cost | 210 |
| Execution Time | 0.0 |
| Nodes Expanded | 620 |
| Score | 300 |

* 1. mediumDottedMaze



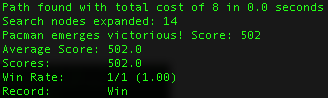
|  |  |
| --- | --- |
| Cost | 1 |
| Execution Time | 0.0 |
| Nodes Expanded | 190 |
| Score | 646 |

* 1. mediumScaryMaze



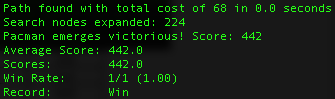
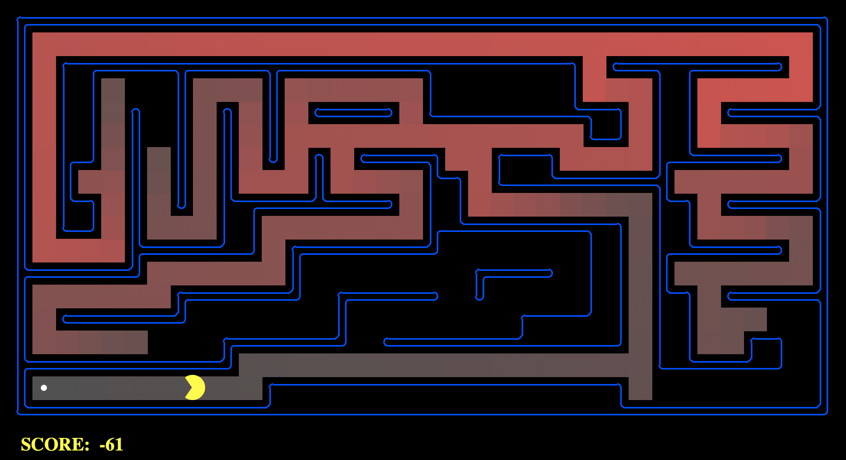
|  |  |
| --- | --- |
| Cost | 68719479864 |
| Execution Time | 0.0 |
| Nodes Expanded | 108 |
| Score | 418 |

1. A\* Search
   1. tinyMaze



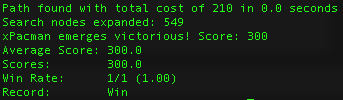
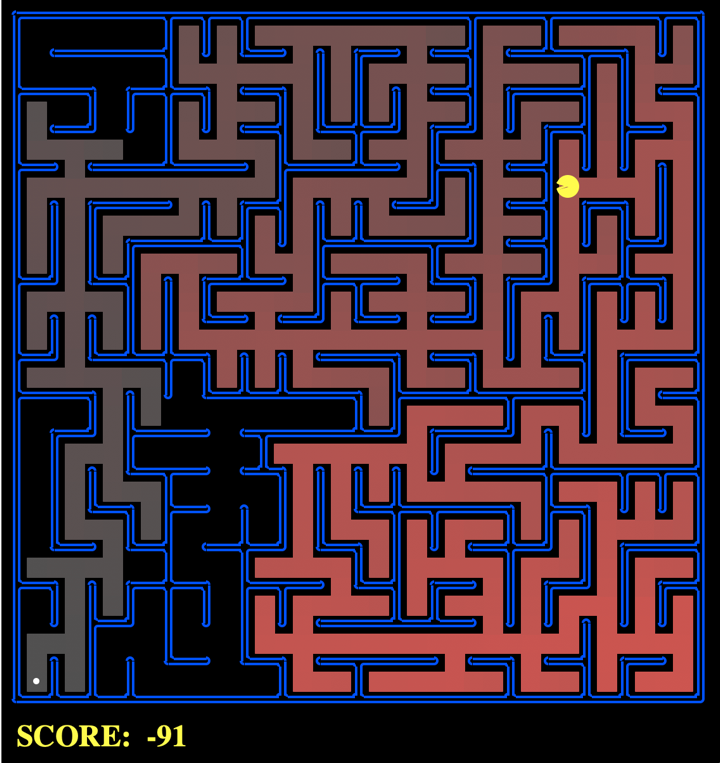
|  |  |
| --- | --- |
| Cost | 8 |
| Execution Time | 0.0 |
| Nodes Expanded | 14 |
| Score | 502 |

* 1. mediumMaze



|  |  |
| --- | --- |
| Cost | 68 |
| Execution Time | 0.0 |
| Nodes Expanded | 224 |
| Score | 442 |

* 1. bigMaze



|  |  |
| --- | --- |
| Cost | 210 |
| Execution Time | 0.0 |
| Nodes Expanded | 549 |
| Score | 380 |

**SUMMARY CHART**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Cost** | **Execution** | **Time Complexity** |
| **DFS** | **tinyMaze** | **10** | **0.0** | **15** |
| **mediumMaze** | **130** | **0.0** | **146** |
| **bigMaze** | **210** | **0.0** | **390** |
| **BFS** | **tinyMaze** | **8** | **0.0** | **16** |
| **mediumMaze** | **68** | **0.0** | **275** |
| **bigMaze** | **210** | **0.0** | **620** |
| **UCS** | **tinyMaze** | **8** | **0.0** | **16** |
| **mediumMaze** | **68** | **0.0** | **275** |
| **bigMaze** | **210** | **0.0** | **620** |
| **A\*** | **tinyMaze** | **8** | **0.0** | **14** |
| **mediumMaze** | **68** | **0.0** | **224** |
| **bigMaze** | **210** | **0.0** | **549** |

**There is a large problem with the algorithms as all the time executions were 0.0 seconds. The analysis of all these algorithms will consist of only cost and time complexity.**

**UCS is similar to BFS because the costs of the actions are “1”. The only time that they would be different is when they are being compared to mediumDottedMaze or mediumScaryMaze which changes the cost of each action.**

**The very first easy thing to compare is the cost of each DFS and BFS in tinyMaze because of the ability to actually count the path. DFS has a cost of 10 while BFS has a cost of 8. This is easily verifiable as the DFS path would take the first node and expand it until it reaches the solution. Due to the algorithm, pacman will expand his left node which causes him 2 extra steps in the cost. For BFS, it is 8 because that is the “depth” of the solution. A\* also has the same cost because BFS and A\* are optimal, but DFS is not. Comparing the time complexity, A\* has the lowest time complexity because of its heuristics. This may be due to the fact that at left side, the heuristics increases due to moving away from the solution. It doesn’t expand those nodes anymore because the other path is more optimal. DFS has the second lowest time complexity because it searches one path and luckily there is a solution. This allows DFS to only have expanded 15 nodes. BFS has the most expanded nodes because it expands both paths at the same time, causing it to have a time complexity of 16 nodes.**

**For mediumMaze the cost for DFS is 130, for BFS, UCS, A\* the cost is 68. The cost for DFS is twice of the algorithms because it expands the path to the left. There is a solution at the end of this path, which is 130 steps. The time complexity for all algorithm varies. DFS has a time complexity of 146 since it is only expanding on a “straight” path. The second longest is A\* with a time complexity of 224. Analyzing the map, it is easy to understand why. It first expands to the left which decreases the distance to the solution. As it approaches the solution, it has to turn around which increases the heuristic every step. After a while the heuristic is finally greater than going down from the spawn point. From this point on, the algorithm expands going down the spawn point and reaching an optimal solution. BFS has the largest number of nodes expanded just because it has to expand all the nodes on the left side even if there is no solution. There is a gap in between the red to show that where it did not expand the nodes due to finding the solution.**

**All the algorithms have the same cost because there is only one path to the solution. The time complexity varies with DFS being the lowest, followed by A\*, and finally BFS and UCS tied for 3rd place. DFS has the lowest time complexity due to how the nodes were expanded. If the nodes were expanded in reverse, it might have expanded more nodes on the right side of the maze. A\* has the second largest number of nodes expanded. It starts to expand to the left, which is represented by the red on the maze. The algorithm realizes that there is no way to reach the solution from expanding to the left, so it backtracks until it is able to find the same path that the DFS algorithm found and expand until it finds the solution. BFS and UCS expands every node on the right side and almost every node on the entire maze until the end, resulting in the largest number of nodes expanded.**