East West University

Lab03 Report

Artificial Intelligence

Course Code: CSE366

Sec: 02

Submitted to

Dr Raihan Ul Islam

Associate Professor

Department of Computer Science and Engineering,

East West University

Submitted by

Name: Tahrima Billal Biva

Id: 2020-3-60-049

Date of Submission: 16.12.24

Topic:BFS,DFS and UCS

```
PS D:\Pacman\search> python pacman.py -l tinyMaze -p SearchAgent -a fn=bfs
readCommand argv {argv}
[SearchAgent] using function bfs
[SearchAgent] using problem type PositionSearchProblem
Execution Time: 0.00196170 seconds
Path found with total cost of 8 in 0.0 seconds
Pacman emerges victorious! Score: 502
Average Score: 502.0
Scores:        502.0
Win Rate:      1/1 (1.00)
Record:        Win
PS D:\Pacman\search> python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs
readCommand argv {argv}
[SearchAgent] using function bfs
[SearchAgent] using problem type PositionSearchProblem
Execution Time: 0.03055760 seconds
Path found with total cost of 68 in 0.0 seconds
Pacman emerges victorious! Score: 442
Average Score: 442.0
Scores:        442.0
Win Rate:      1/1 (1.00)
Record:        Win
PS D:\Pacman\search> python pacman.py -l bigMaze -p SearchAgent -a fn=bfs
readCommand argv {argv}
[SearchAgent] using function bfs
[SearchAgent] using problem type PositionSearchProblem
Execution Time: 0.05929830 seconds
Path found with total cost of 210 in 0.1 seconds
Pacman emerges victorious! Score: 300
Average Score: 300.0
Scores:        300.0
Win Rate:      1/1 (1.00)
Record:        Win
```

Figure 1: BFS Execution of All Mazes

```
PS D:\Pacman\search> python pacman.py -l tinyMaze -p SearchAgent -a fn=dfs
readCommand argv {argv}
[SearchAgent] using function dfs
[SearchAgent] using problem type PositionSearchProblem
Execution Time: 0.00162290 seconds
Path found with total cost of 10 in 0.0 seconds
Pacman emerges victorious! Score: 500
Average Score: 500.0
Scores:        500.0
Win Rate:      1/1 (1.00)
Record:        Win
PS D:\Pacman\search> python pacman.py -l mediumMaze -p SearchAgent -a fn=dfs
readCommand argv {argv}
[SearchAgent] using function dfs
[SearchAgent] using problem type PositionSearchProblem
Execution Time: 0.01317920 seconds
Path found with total cost of 130 in 0.0 seconds
Pacman emerges victorious! Score: 380
Average Score: 380.0
Scores:        380.0
Win Rate:      1/1 (1.00)
Record:        Win
PS D:\Pacman\search> python pacman.py -l bigMaze -p SearchAgent -a fn=dfs
readCommand argv {argv}
[SearchAgent] using function dfs
[SearchAgent] using problem type PositionSearchProblem
Execution Time: 0.03551580 seconds
Path found with total cost of 210 in 0.0 seconds
Pacman emerges victorious! Score: 300
Average Score: 300.0
Scores:        300.0
Win Rate:      1/1 (1.00)
Record:        Win
```

Figure 2: DFS Execution of All Mazes

```
PS D:\Pacman\search> python pacman.py -l tinyMaze -p SearchAgent -a fn=ucs
readCommand argv {argv}
[SearchAgent] using function ucs
[SearchAgent] using problem type PositionSearchProblem
Execution Time: 0.00224320 seconds
Path found with total cost of 8 in 0.0 seconds
Pacman emerges victorious! Score: 502
Average Score: 502.0
Scores:        502.0
Win Rate:      1/1 (1.00)
Record:        Win
PS D:\Pacman\search> python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
readCommand argv {argv}
[SearchAgent] using function ucs
[SearchAgent] using problem type PositionSearchProblem
Execution Time: 0.01779590 seconds
Path found with total cost of 68 in 0.0 seconds
Search nodes expanded: 269
Pacman emerges victorious! Score: 442
Average Score: 442.0
Scores:        442.0
Win Rate:      1/1 (1.00)
Record:        Win
PS D:\Pacman\search> python pacman.py -l bigMaze -p SearchAgent -a fn=ucs
readCommand argv {argv}
[SearchAgent] using function ucs
[SearchAgent] using problem type PositionSearchProblem
Execution Time: 0.03768800 seconds
Path found with total cost of 210 in 0.0 seconds
Search nodes expanded: 620
Pacman emerges victorious! Score: 300
Average Score: 300.0
Scores:        300.0
Win Rate:      1/1 (1.00)
Record:        Win
```

Figure 3: UCS Execution of All Mazes

Algorithms	Mazes	Path Cost	Nodes/ Path Expanded	Execution Time (Seconds)	Total Time (Sec)
BFS	tinyMaze	8	15	0.00196170	0.0
BFS	mediumMaze	68	269	0.03055760	0.0
BFS	bigMaze	210	620	0.05929830	0.1
DFS	tinyMaze	10	15	0.00162290	0.0
DFS	mediumMaze	130	146	0.01317920	0.0
DFS	bigMaze	210	390	0.03551580	0.0
UCS	tinyMaze	8	15	0.00224320	0.0
UCS	mediumMaze	68	269	0.01779590	0.0
UCS	bigMaze	210	620	0.03768800	0.0

Here, Execution time defines measuring the total time spent executing the specific search algorithm .And , the total time defines the additional time spent for the whole pacman game ,such as,loading , initializing , rendering graphics and processing the path etc.Nodes/Path Expanded defines number of nodes it has visited .Path cost define the total cost of the solution path.

Comparison:

For tinyMaze, the path cost of dfs is 10 while the path cost of bfs and ucs are 8 .All three algorithms have the same number of nodes expanded which is 15 for tiny maze .Also, the execution time of dfs is lower than the other two algorithms.That means dfs is faster than the other algorithms because it visits less nodes than the other two.

For mediumMaze, the path cost of dfs is 130 while the path cost of bfs and ucs are 68 .The expanded nodes of dfs is 146 while the expanded nodes of bfs and ucs are 269 for medium maze .That means the expanded nodes of dfs is lower than the other two algorithms. Also, the execution time of dfs is lower than the other two algorithms and the execution time of bfs is higher than the other two algorithms.That means dfs is faster than the other algorithms since it visits fewer nodes.

For bigMaze,All three algorithms have the same amount of path cost which is 210.The expanded nodes of dfs is 390 while the expanded nodes of bfs and ucs are 620 for big maze.Also, the execution time of dfs is lower than the other two algorithms and the execution time of bfs is higher than the other two algorithms.That means dfs is faster than the other algorithms because it visits less nodes than the other two.

In conclusion ,dfs is best based on the path costs,expanded nodes and execution times among all three algorithms for all mazes.