RADEME

600.335/435 Artificial Intelligence---Homework 2

Bo Lei

Instructions for running program

•***Complete question 1, the depth-first search. Provide your solution for tinyMaze, mediumMaze, bigMaze, and openMaze.***

In this question, I just modified the *search.py* file and add my code within function *depthFirstSearch(problem)*. The comments to explain my code can be seen in the file.

By running the following 4 commands, we can see the solutions for tinyMaze, mediumMaze, bigMaze and openMaze:

```
python pacman.py -l tinyMaze -p SearchAgent
python pacman.py -l mediumMaze -p SearchAgent
python pacman.py -l bigMaze -z .5 -p SearchAgent
python pacman.py -l openMaze -z .5 -p SearchAgent
```

•***Complete question 2, the breadth-first search. Provide your solution for tinyMaze, mediumMaze, bigMaze, and openMaze***.

In this question, I just modified the *search.py* file and add my code within function *breadthFirstSearch (problem)*. The comments to explain my code can be seen in the file.

By running the following 4 commands, we can see the solutions for tinyMaze, mediumMaze, bigMaze and openMaze:

```
python pacman.py -l tinyMaze -p SearchAgent -a fn=bfs
python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs
python pacman.py -l bigMaze -p SearchAgent -a fn=bfs
python pacman.py -l openMaze -p SearchAgent -a fn=bfs
```

•***Grad Students: In addition to the two previously-mentioned search algorithms, please implement an iterative deepening search. Provide your solution for tinyMaze, mediumMaze, bigMaze, and openMaze.***

In this question, I just modified the *search.py* file and defined a new function *iterativeDeepeningSearch(problem)*. The comments to explain my code can be seen in the file. Also, add abbreviation *ids = iterativeDeepeningSearch.*

By running the following 4 commands, we can see the solutions for tinyMaze, mediumMaze, bigMaze and openMaze:

```
python pacman.py -l tinyMaze -p SearchAgent -a fn=ids
```

```
python pacman.py -l mediumMaze -p SearchAgent -a fn=ids -z .5
python pacman.py -l bigMaze -p SearchAgent -a fn=ids -z .5
python pacman.py -l openMaze -p SearchAgent -a fn=ids -z .5
```

•***Complete question 3, which varies the cost function of your breadth-first search. Provide your solution for mediumMaze with the UCS agent, mediumDotted-Maze with StayEastSearchAgent, and mediumScaryMaze with StayWestSearchAgent.***

In this question, I just modified the *search.py* file and add my code within function *uniformCostSearch(problem)*. The comments to explain my code can be seen in the file.

By running the following 3 commands, we can see the solutions for mediumMaze with the UCS agent, mediumDotted-Maze with StayEastSearchAgent, and mediumScaryMaze with StayWestSearchAgent.

```
python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
python pacman.py -l mediumDottedMaze -p StayEastSearchAgent
python pacman.py -l mediumScaryMaze -p StayWestSearchAgent
```

•***Complete question 4, the A* search using manhattanHeuristic. Provide your solution for tinyMaze, mediumMaze, bigMaze, and openMaze.***

In this question, I just modified the *search.py* file and add my code within function *uniformCostSearch(problem)*. The comments to explain my code can be seen in the file.

By running the following 4 commands, we can see the solutions for tinyMaze, mediumMaze, bigMaze and openMaze:

```
python pacman.py -l tinyMaze -z .5 -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic
python pacman.py -l mediumMaze -z .5 -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic
python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic
python pacman.py -l openMaze -z .5 -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic
```

•***Complete question 5, which solves the corners problem with a BFS agent. Provide your solution for tinyCorners and mediumCorners***.

In this question, I just modified the *searchAgents.py* file and add my code within each function of class *CornersProblem(search.SearchProblem)* . The comments to explain my code can be seen in the file.

By running the following 2 commands, we can see the solutions for tinyCorners and mediumCorners:

```
python pacman.py -l tinyCorners -p SearchAgent -a fn=bfs,prob=CornersProblem
python pacman.py -l mediumCorners -p SearchAgent -a fn=bfs,prob=CornersProblem
```

•***Grad Students: Complete question 6, which creates a new heuristic for the corners problem. Provide your solution for tinyCorners and mediumCorners.***

In this question, I just modified the *searchAgents.py* file and add my code within function *cornersHeuristic(state, problem)*. The comments to explain my code can be seen in the file.

By running the following 2 commands, we can see the solutions for tinyCorners and mediumCorners:

```
python pacman.py -l tinyCorners -p AStarCornersAgent -z 0.5
python pacman.py -l mediumCorners -p AStarCornersAgent -z 0.5
```

•***Complete question 7, which solves the eating all the dots problem with A* with a null heuristic. Provide your solution for testSearch and trickySearch.***

In this question, we just need to let function *foodHeuristic(state, problem)* return 0 and test it. In this way, it is a null heuristic.

```
python pacman.py -l testSearch -p AStarFoodSearchAgent
python pacman.py -l trickySearch -p AStarFoodSearchAgent
```

•***Grad Students: Implement foodHeuristic and provide your solutions using this new heuristic for testSearch and trickySearch.***

In this question, I just modified the *searchAgents.py* file and add my code within function *foodHeuristic(state, problem)*. The comments to explain my code can be seen in the file.

The commands are the same as the ones in last problem.