CS-361L Artificial Intelligence Lab 05

Type of Lab: Open Ended **Weightage: 5%**

CLO 1: Apply Informed and **Informed** Search Techniques and build the ability to theoretical and practical understanding of Blind and Informed machine search and machine learning techniques.

Student implements Informed (**A* and Heuristic**) techniques to search goals for Pacman	**Cognitive/Understanding**	CLO1	Rubric A
Demonstrate ethical and professional responsibilities involved in completion of Tasks	**Affective/Valuing**	(CLO6)	Rubric B

Reference: The Lab contents are extracted from the BerkeleyX/CS188

Rubric A: Cognitive Domain

Evaluation Method: GA shall evaluate the students for Questions according to following rubrics.

CLO	**0**	**1**	**2**	**3**	**4**
CLO1	Student was not able to understand challenge 01 (A*)	Student was able to partially complete challenge 01 (A* Implementation)	Student was not able to complete challenge 01 (A* Implementation)	Student was not able to complete challenge 02	Student was not able to complete challenge 03
Roll Number					

Reference: The Lab contents are extracted from the BerkeleyX/CS188

Rubric B: Affective Domain: Lab Staff shall help GA in evaluation of the students for their CLO 6

CLO 6	**0**	**1**	**2**	**3**
Demonstrate ethical and professional responsibilities involved in completion of Tasks.	Student was not on time in the lab	Student showed some unethical behavior and was late in lab	Student was on time and showed some unethical behavior	Student was obedient and showed ethical behavior
Roll Number				

Reference: The Lab contents are extracted from the BerkeleyX/CS188

Lab 05: Implementation of A* and Heuristic Search

Note for Instructor: Student should start with solution of UCS

Challenge 01: A* and Heuristic Search

Implement A* graph search in the empty function aStarSearch in search.py. A* takes a heuristic function as an argument. Heuristics take two arguments: a state in the search problem (the main argument), and the problem itself (for reference information). The nullHeuristic heuristic function in search.py is a trivial example.

Step1:

Open the search.py file and copy your UCS code into the `aStarSearch` function.

Step2:

You can test your A* implementation on the original problem of finding a path through a maze to a fixed position using the Manhattan distance heuristic. Write to Manhattan Heuristic function and update the cost calculation formula. If there are two points A (x1,y1) and B(x2,y2) in xy-coordinate space, the Manhattan distance can be calculated as.

```
Distance= abs(x2-x1) + abs(y2-y1)
```

Hint (you can get goal state information using **problem.goal**)

Define the function in search.py and write the code of manHattanHeuristic in it.

```
def manHattanHeuristic(state, problem=None):
```

Step 3:

Modify aStarSearch function such that it shall consider the cost of heuristic as well as cost from the parent node. For heuristic, call the method that you have created in step 1.

Before running the code, you should confirm following abbreviation at the end of the search.py is added. If not, then add it

```
astar = aStarSearch
```

Run the aStar algorithm using following command

```
python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar
```

You should see that A* finds the optimal solution slightly faster than uniform cost search (about less search nodes expanded in our implementation, but ties in priority may make your numbers differ slightly).

Reference: The Lab contents are extracted from the BerkeleyX/CS188

Challenge 02:

1. Fill following table with the information.

	A* Heuristic	UCS
Total Cost		
Nodes Expanded		
Score		

2. **Why Node Expanded is Greater in UCS and Less in A* Heuristic?**

3. **Compare other parameters and give reason why they greater/less/equal.**

Challenge 03: Write a function that calculates heuristic based on the Euclidean distance.

Now run the code on mediumClass maze for UCS, Manhattan and Euclidean distance one by one and report your results. Also discuss the results why these results are coming.

```
Python pacman.py –l mediumClassicMaze –p SearchAgent –a fn=astar
```

Reference: The Lab contents are extracted from the BerkeleyX/CS188

	Euclidean	Manhattan	UCS
Total Cost			
Nodes Expanded			
Score			

Reference: The Lab contents are extracted from the BerkeleyX/CS188