Koç University
COMP341
Introduction to Artificial Intelligence
Assignment 1

Instructor: Barış Akgün
Due Date: October 15 2018, 23:59
Submission Through: Blackboard
Make sure you read and understand every part of this document

This programming assignment will test your knowledge and your implementation abilities of what you have learned in the uninformed and informed search parts of the class. You are asked to complete a coding part and answer a few questions about how it runs. The coding part of the homework will follow the Berkeley CS188 Spring 2014 pacman project P1: Search at `http://ai.berkeley.edu/search.html`. The questions for the report part are given in this document.

This homework must be completed individually. Discussion about algorithms, algorithm properties, code structure, and Python is allowed but group work is not. Coming up with the same approach and talking about the ways of implementation leads to very similar code which is treated as plagiarism! Furthermore, do not discuss the answers directly as it will lead to similar sentences which is treated as plagiarism. If you are unsure, you should not discuss. Any academic dishonesty, will not be tolerated. **By submitting your assignment, you agree to abide by the Koç University codes of conduct.**

You may find yourself having trouble implementing the coding part. In this case, we are going to let you use someone else's code to answer the given questions, as long as you credit the person or the website you take the code from. If you chose this option, we are only going to grade your report.

Grading

You are given two options about submitting your homework: (1) both code and report, and (2) only report. The second option is given to you in case you are not able to implement the programming part. These options will be graded differently:

- **Code and Report:** The code part and the report will have 2:1 weight ratio in your submission (programming 2/3, report 1/3).

- **Only Report:** The report part will be treated as 1/2 of the total grade. You **must** credit the code you used and **must not** submit a code part.

The solution code for the homeworks can be found online. We are going to compare your submission with these sources. We are also going to compare your code to previous submissions of Koç students. If your codes similarity level is above a certain threshold, your code will be scrutinized. If we see any plagiarism, you will lose points in the best case and disciplinary action will be taken against you in the worst.

Part 1: Programming

You are going to do the 8 programming questions about search given here: `http://ai.berkeley.edu/search.html`. You are only required to change *search.py* and *searchAgents.py*. If you have any issues with other parts of the code let your instructor or TA know ASAP, even if you manage to solve your problem. Use the data structures in *util.py* for the autograder to work properly. If the you think you have the right answer but the autograder is not giving you any points, try to run it on individual questions (look at P0 for details on how to use the *autograder.py*). The file *autograder.py* is missing an `import pprint`, which might cause issues in some setups.

Hints

- We understand that having too many files to go through might be troublesome, however keep in mind that **you do not need to go over all the files**. Read the P1:Search documentation in the Berkeley Website and the comment sections of the relevant files/functions that you are going to work on. If you start worrying about implementation details of Pacman, you will get lost and lose track of how to handle the assignment requirements.

- **Always** read the comment sections of the given code before starting your implementation. These comments are very useful and they will save you precious time later on.

Part 2: Report

This part includes answering the following questions based on your programs output on the given pacman tests. Look specfically at *bigMaze* and the *openMaze* for stark comparisons.

You are expected to answer the questions concisely. Five sentences is more than enough for most of them. Limit yourself to 200 words. It is okay if you over-generalize, as long as your direction is clear and correct. Note that some answers are already in the provided link to the Berkeley site. **Answers that are longer than 200 words may not be graded!**

Create a PDF file named *report.pdf* containing your answers for submission.

Written Q1:
What are some differences between DFS and BFS in terms of path cost and number of expanded nodes? When and why would you prefer BFS over DFS? When and why would you prefer DFS over BFS?

Written Q2:
What are some differences between UCS and A* in terms of path cost and number of expanded nodes? When and why would you prefer UCS over A*? When and why would you prefer A* over UCS?

Written Q3:
Comment on your choice of state in the four corners problem. Why does it allow you to solve the problem?

Written Q4:
Comment on your choice of heuristic in the four corners problem. Why did you settle on that heuristic? Why is it admissible and consistent?

Written Q5:
Comment on your choice of heuristic in the eating all the dots problem. Why did you settle on that heuristic? Why is it admissible and consistent?

Written Q6:
What are some practical differences between a consistent and an inadmissible heuristic, in terms of path cost and number of expanded nodes? When and why would you prefer an inadmissible heuristic over a consistent one? When and why would you prefer a consistent heuristic over an inadmissible one?

Submission

You are going to submit a compressed archive through the blackboard site. The file should extract to a folder with your student ID without the leading zeros. This folder should only contain *report.pdf*, *search.py* and *searchAgents.py*. Other files will be deleted and/or overwritten. Do not submit any code if you only want us to grade your report.

Important: Download your submission to make sure it is not corrupted and it has your latest report/code. You are only going to be graded by your blackboard submission.

Submission Instructions

- You are going to submit a compressed archive through the blackboard site. The file can have *zip*, *rar*, *tar*, *rar*, *tar.gz* or *7z* format.

- This compressed file should extract to a folder with your student identification number with the two leading zeros removed which should have 5 digits. Multiple folders (apart from operating system ones such as MACOSX or DS Store) greatly slows us down and as such will result in penalties

- Code that does not run or that does not terminate will not receive any credits.

- Do not trust the way that your operating system extracts your file. They will mostly put the contents inside a folder with the same name as the compressed file. We are going to call a program (based on your file extension) from the command line. The safest way is to put everyting inside a folder with your ID, then compress the folder and give it your ID as its name.

- Once you are sure about your assignment and the compressed file, submit it through Blackboard.

- After you submit your code, download it and check if it is the one you intended to submit.

- **DO NOT SUBMIT CODE THAT DOES NOT TERMINATE OR THAT BLOWS UP THE MEMORY.**

Let us know if you need any help with setting up your compressed file. This is very important. We will put all of your compressed files into a folder and run multiple scripts to extract, clean up, grade and do plagiarism checking. If you do not follow the above instructions, then scripts might fail. If these scripts fail then you will receive a 0 from the coding part and the report will be worth 1/3 of your grade.