Homework #2

Due Feb 19 by 6pm **Points** 100 **Submitting** a file upload
File Types pdf, zip, txt, and py **Available** until Mar 1 at 6pm

This assignment was locked Mar 1 at 6pm.

This homework will be graded out of 100 points. It will count 6% of the grade. The homework has to be done individually, no collaboration with other student and no copying answers from anyone/anywhere. Homeworks are due on Friday at 6:00pm. Late homeworks will lose 10% of the maximum total points for each day they are late. Weekends do not count, so if you submit by 6:00pm on Monday you lose only 10%.

Submit the assignment electronically using canvas. The written part has to be uploaded as a .pdf file. Please do not submit word files! Please also avoid submitting handwritten answers, unless your handwriting is very easy to read and the image is clear.

Written part (80 points)

1. [20 points] The functions *g* and *h* each play a different role in A*. What are those roles? What happens when you emphasize or de-emphasize one of them by using different weights in *f(n)*? Consider the case in which *f(n) = (1 - w)g(n) + wh(n)*, with *0 <= w <=1*. This weighted A* search is described briefly in the the textbook (4th edition, sect 3.5.4). Be specific and analyze what happens for different values of *w.*
2. [20 points] You want to reduce the memory used by A*. You come up with the following idea: you keep in the queue only the N best nodes (i.e, the nodes with lower costs), for some positive value of N. When the queue is full and a new node has to be stored, the worst node is deleted from the queue and removed from consideration.
   1. If an admissible heuristic is used, will the modified algorithm find the optimal solution? Explain why (or why not). Are there any additional constraints?
   2. If a perfect heuristic is used, i.e. *for all n h(n)=h*(n)* , where *h*(n)* is the cost of the optimal path from *n* to goal, will the modified algorithm find the optimal solution? Explain.
3. [10 points] Answer the following questions on Uniform Cost search briefly but precisely:
   1. Is it possible for Uniform Cost to expand more nodes than Breadth-First search? Feel free to use an example to support your answer.
   2. Does Uniform Cost search expand more nodes than A*? Why (or why not)?
4. [10 points] Does the fact that A* is "optimally efficient" mean that A* will never expand more nodes than any other algorithm?
5. [20 points] Inspired by the iterative deepening algorithm, you decide to design an "iterative broadening algorithm". The idea is to start with 2 children, and do depth-first search limiting at each node expansion the number of children to 2. If you fail to find a solution, you restart the search from the beginning increasing the number of children by 1. Repeat this process until you find a solution.
   1. What advantages, if any, do you see in this algorithm? What shortcomings?
   2. For what type of search spaces do you think this algorithm will be useful?

Programming part (20 points) --- This part is now finalized

For this programming part we will use the Romanian map shown in the textbook and already defined in the aima code. You can use the notebook from the previous assignment, adding the call to the compare_searchers function. For the second question, you need to modify the compare_searchers function to include the path cost. Submit the code you wrote and the results you got.

1. [10 points] Compare the performance (using the function compare_searchers) of the following search algorithms:
   1. breadth_first_tree_search,
   2. breadth_first_graph_search,
   3. depth_first_graph_search,
   4. uniform_cost_search
   5. astar_search
      to solve these problems:
   6. find a path from Arad to Bucharest
   7. find a path from Sibiu to Bucharest
   8. find a path from Eforie to Timisoara
2. [10 points] Add to the compare function above the cost of each solution to a problem for each of the algorithms. This requires reading the code to find out where the path cost is saved and to modify the compare_searchers function to include the path cost.

hw2-rubric

Criteria	Ratings							Pts
Question1 Graded by Ben	**20 pts** **Full Marks**	**20 to >15.0 pts** **Minor Issues** Insufficient explanation or minor issues with correctness.	**15 to >10.0 pts** **Major issues** Missing part of the question. Partially incorrect response.	**10 to >0 pts** **Incorrect** Response is incorrect, unrelated to the question, or missing.				20 pts
Question2 Graded by Fei	**20 pts** **Full Marks**	**20 to >15.0 pts** **Need more explanations**	**15 to >10.0 pts** **Wrong answer in either 2.1 or 2.2**	**10 to >0.0 pts** **Wrong answer in both 2.1 and 2.2**	**0 pts** **No Marks**			20 pts
Question3 Graded by Robert	**10 pts** **Full Marks**	**8 pts** **Minor Issues** Lacks justification, or partially incorrect.	**5 pts** **Major Issues** One or more incorrect answers.	**2 pts** **Incorrect** Incorrect answers, but some attempt was made.	**0 pts** **No Marks**			10 pts
Question4 Graded by Ioanna Answer (yes/no): 3 points explanation: 7 points	**10 pts** **Full Marks**	**8 pts** **correct answer, minor mistakes or missing points in explanation**	**7 pts** **incorrect answer but good explanation/ correct answer but explanation is missing critical points**	**5 pts** **incorrect answer and not sufficient explanation**	**3 pts** **correct answer, no explanation**	**2 pts** **incorrect answer and no explanation**	**0 pts** **No Marks**	10 pts
Question5 Graded by Trevor	**20 pts** **Full Marks**	**15 pts** **Minor issues** Minor issues with reasoning with one or more parts of the question.	**10 pts** **Some major issues** Major issues or fundamental misunderstanding with what was being asked with one or more question	**5 pts** **Fundamental misunderstanding** major issues with reasoning about both sections.	**0 pts** **No Marks**			20 pts

Criteria	Ratings			Pts
Programming Graded by Robert	**20 pts** **Full Marks**	**10 pts** **Missing cost of solution**	**0 pts** **No Marks**	20 pts

Total Points: 100