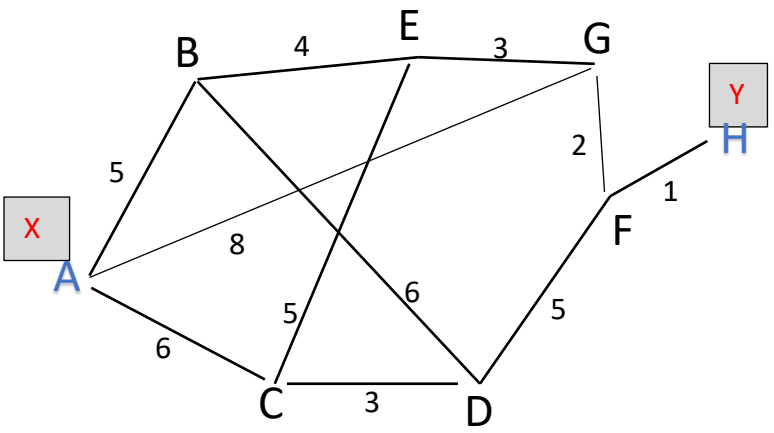
Artificial Intelligence – Spring 2022 – Assignment 2

Objective: this

- Becoming proficient with problem representation and search strategies
- Get familiar with the development of heuristics

Suppose that we have a map consisting of cities A, B, …, H with the road and distances between cities (whenever they are connected) as shown in the above figure. Two agents, one (X) is at A and another (Y) is at H. They want to meet each other at some cities. Assume that the agents can only move to the neighboring cities, one at a time. At every step, each must move to a neighboring city, i.e., they cannot just stay in one place to wait for the other agent go come. Furthermore, we assume that each move taking one time step, i.e., the distance does not affect how fast/slow an agent moves.



As an example, they can meet each other after two steps: X moves to B then D and Y moves to F then D. However, if X moves to G and Y moves to F in the first step, then they cannot meet each other after two steps because X must move to A, E, or F while Y must move to G, D, or H.

Let us consider two different definitions of the cost of a solution to the above problem:

- it is defined as the number of steps that the agents must move to meet each other; in the example, X -> B -> D and Y -> F -> D the cost is 2; or
- it is defined as the total distance that the agents must travel meet each other; in the example, X -> B -> D and Y -> F -> D the cost is 7 = 11 (for X) + 6 (for Y);

Following the first definition, the solution that requires X -> B -> D and Y -> F -> D is an optimal one. However, it is not optimal according to the second definition since X -> C -> D and Y -> F -> D has a lower cost (15).

In Assignment 1, you have developed a representation for the problem as a search problem. In this assignment, you need to use develop the code to compute the solution of the problem. You can use the code from the textbook website (search.py or java code – it is organized differently from the python code, but the code is available) or you can develop your own code. For all the experiments, let us use the code that removes visited states.

Create a comparison of the following search strategies: breadth first search, depth first search, iterative deepening search, and A* search. Specifically, for each strategy, submit a printout of

- the number of iterations that the algorithm must run before it completes;
- for each iteration, the selected node and the nodes that are added to the frontier; and
- the final solution.

For the A* search, do run it with two different costs as defined above. So, you need to make five calls to different procedures.

For the A* search, you will need to invent some heuristic. One possible heuristic is to estimate the straight distance between nodes by taking a reasonable number that can easily derived from the above map (e.g., the straight distance from A to F could be 9 since should be smaller than the path AGF (10); etc). If you plan to do it, complete the following table. I did add a few more numbers (in red).

	A	B	C	D	E	F	G	H
A	0	5	6	7	6	9	8	10
B		0						
C			0					
D				0				
E					0			
						0		
G							0	
H								0

Submit your written homework and codes on Canvas by 11:59 pm, February 17, 2022. Do include a README file that explains how your conde should be compiled and ran.

Grading:

- README file: 5 points
- For each strategy, correct output and solution, 18 points (BFS, DFS, IDS, A* with cost #1 and A* with cost #2)
- Heuristic for A*: 5 points