

Week 2 - Search Algorithms

MACHINE INTELLIGENCE LABORATORY- PESU UE19CS305

Teaching Assistants-

vighneshkamath43@gmail.com

sarasharish2000@gmail.com



Search Algorithms aim at navigating from a start state to a goal state by transitioning through intermediate states. It also consists of a state space which is a set of all possible states where you can be.

There are many informed and uninformed search algorithms that exist and are very popular.

A* search, Uniform Cost search (UCS), Depth First Search (DFS), Greedy Search to name a few.

In this assignment you are required to write two functions A_star_Traversal and DFS_Traversal which implements the respective algorithm

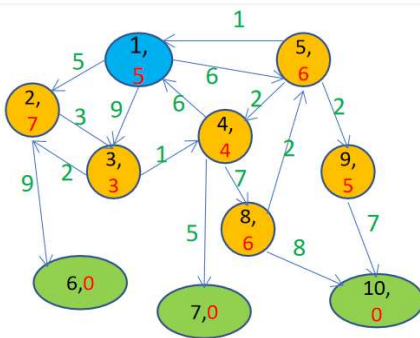
Your task is to complete the code for this function.

You are provided with the following files:

1. week2.py
2. SampleTest.py

Note: These sample test cases are just for your reference.

SAMPLE TEST CASE GRAPH



Numbers in Black represent Node numbers, numbers in Green represent Cost and numbers in Red represent heuristic values from that node
Start Node is in Blue
Goal Nodes are in Green

Note: This is the same graph that was used in class for A* explanation hence you must be easily able to trace the answer

Things to note with respect to the algorithm:

1. Search Algorithms **don't** travel entire space, and find all paths and choose the optimal
2. Algorithms are refined to reach the goal at using best path in the first run itself.
3. The path it returns may/may not be equal to the explored node data structure.

Important Points:

1. Please do not make changes to the function definitions that are provided to you. Use the skeleton as it has been given. Also do not make changes to the sample test file provided to you. Run it as it is.
2. You are free to write any helper functions that can be called in any of these predefined functions given to you.
3. **Your code will be auto evaluated by our testing script and our dataset and test cases will not be revealed. Please ensure you take care of all edge cases!**
4. **In case of ambiguity, pick nodes in lexicographical order.**
5. Please adhere to the above point strictly failing may result in wrong answer. (example from node A you can go to B C D, then choose to explore B at first. This example is not wrt to any algorithm and just general point)
6. **The experiment is subjected to zero tolerance for plagiarism. Your code will be tested for plagiarism against every code of all the sections and if found plagiarized both the receiver and provider will get zero marks without any scope for explanation.**

7. **Kindly do not change variables name or other technique to escape from plagiarism, as the plagiarism checker is able to catch such plagiarism.**
8. **Implement the algorithm in the right way, using appropriate data structures. Inefficient code may lead to Time Limit Exceeded error.**
9. Hidden test cases will not be revealed post evaluation.

week2 .py

Contains two function

1. DFS_Traversal
2. A_star_Traversal

INPUT_PARAMETER			
Name	Description	Type	Example
cost	Cost matrix. Cost of moving from node i to node j is defined by cost[i][j]	list of (list of integer/float)	[[0,1,2.1] [1,0,1] [3.1,1,0]]
heuristic	Heuristic data structure for each node, heuristic at node I is defined by heuristic[I]	List of integer/float	[1,2.1,0]
start_point	The start point where the algo starts	integer	0
goals	List of goal nodes,length==number of goals	List of integer	[2,3]

RETURN_PARAMETER			
Name	Description	Type	Example
path	The path should contain the path that the algorithm returns	List of integer	[1,2,3]

1. **You are required to return paths using the respective algorithm**
2. **You may write your own helper function if needed**
3. **You can import libraries that come built-in with python 3.7**
4. **You cannot change the skeleton of the code**

SampleTest.py

1. This will help you check your code.
2. Note that the test case used in this is same as the graph defined above
3. **Passing the cases in this does not ensure full marks ,you will need to take care of edge cases**
4. Name your code file as YOUR_SRN.py
5. Run the command **python3 SampleTest.py --SRN YOUR_SRN (incase of any import error use the below command)**

python3.7 SampleTest.py --SRN YOUR_SRN

SUBMISSION FORMAT and DEADLINE:

You are required to complete code in week2.py and **rename it to SRN.py**

Failing to submit in the right format will lead to zero marks without a chance for correction

Example: If your SRN is PES1201801819 your file should be PES1201801819.py

If your SRN is PES1UG19CS020 your file should be PES1UG19CS020.py

Use your SRN and not USN

All the helper function should be in the same file, make sure you run the sample test case before submitting.

Deadline: on or before 05/09/2021 10:00PM

Where to submit: Edmodo