**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

Batch No. :

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS**

**Artificial Intelligence (BITS F444/ CS F407)**

**I Semester 2017-18**

**Programming Assignment-1**

**Coding Details**

**(September 14, 2017)**

1. **ID :** 2015A7PS0078P

**Name:** Naveen Venkat

1. **Names of Submitted files :**
   1. structures.py
   2. driver.py
2. Total number of submitted files: **2**
3. Name of the folder : **Assignment 1**
4. Have you checked that all the files you are submitting have your name in the top? **yes**
5. Have you checked that all the files you are submitting are in the folder as specified in 4 (and no subfolder exists)? **yes**
6. **Problem formulation**
   1. **State representation:** A tuple of two entities  
      **I)** a boolean list of dirt in cells (True means dirt is present in the corresponding cell). The boolean list is visualized as the row major representation of the 2D world.  
      **II)** a 2-tuple of x & y positions of the agent when the environment is visualized as a 2D matrix ( (0,0) on the top left and (n-1, n-1) on the bottom right )
   2. **How is the Initial state generated?** Initial position of the agent would be in randomly generated in one of the four corners, and the dirt will be generated randomly in ‘p’ fraction of the total cells.
   3. **What is the goal state?** A goal state in this scenario would have no dirt in any of the cells and the agent would be in one of the 4 resting positions as specified.
   4. **Are there more than one goal states?** Yes
   5. **If yes, then describe all the goal states:** There are 4 possible goal states, one corresponding to each of the 4 resting positions of the agent. The dirt in all these states would be absent in any cell.

eg. For a 2x2 world, goal states are  
[ [False, False, False, False], [0,0] ]  
[ [False, False, False, False], [0,1] ]

[ [False, False, False, False], [1,0] ]

[ [False, False, False, False], [1,1] ]

* 1. **State representation in Python:** Represented as a **List** of two entities as described in part (a)   
     eg. For a 2x2 world, a possible state could be: [ [True, False, False, True], [0,1] ]

1. **Successor function description :** This function would return the next state by applying the given action on the given state
2. **Uninformed Search Technique (T1) details**
   1. Technique used for search: **Breadth First Search**

* 1. Reason for selecting this technique over the other two: **For analyzing how the speed of execution depends on the memory constraints**
  2. Is the search applied on tiles or on states?: **States**
  3. Error handling and reporting (yes/No): **No**
  4. List the errors handled: **N/A**
  5. Data Structure description for the tree node (in maximum two lines): Each node contains **state, link to parent node,** and **action** taken to reach the state
  6. Code status (implemented fully/ partially/ not done): **Partially implemented**

1. **Informed search Technique (T2) details:**
   1. Technique used for search: N/A
   2. Reason for selecting this technique over others: N/A
   3. Does this technique look at a tile? N/A
   4. Does this technique use a state? N/A
   5. Code status (implemented fully/ partially/ not done): Not done
   6. Define the heuristics (in words) used in your program
      1. h1 : N/A
      2. h2 : N/A
   7. Compute the heuristic values for the following windows with dirt in the neighborhood of the centre position of the vacuum cleaner.
      1. h1 = , h2 =
      2. h1 = , h2 =
      3. h1 = , h2 =
      4. h1 = , h2 =

(i)

(ii)

(iii)

(iv)

1. GUI details
   1. Created the GUI (yes/ N0): **No**
   2. Have created it according to the specifications? **No**
   3. Which module of Python used for creating graphics? **N/A**
   4. Is this under the standard Python library or not? **N/A**
   5. If not, why? **N/A**
   6. Are the window panes working independently? **N/A**
2. Graphics details:
   1. Is turtle graphics working fine for movement of the intelligent vacuum cleaner? **N/A**
   2. How are you creating the room tiles? **N/A**
   3. How are you showing the dirt? **N/A**
   4. How are you showing the resting position of the vacuum cleaner? **N/A**
   5. Are you showing the movement of the vacuum cleaner (turtle cursor) as the execution of T1 goes on? Why? **N/A**
   6. Are you showing the movement of the vacuum cleaner (turtle cursor) as the execution of T2 goes on? Why? **N/A**
3. Compilation Details:
   1. Code Compiles (Yes/ No): **Yes**
   2. Mention the .py files that do not compile: **N/A**
   3. Any specific function that does not compile: **N/A**
   4. Ensured the compatibility of your code with the specified Python version: **Yes**
   5. Instructions for compilation of your files mentioning the multi file compilation process used by you (We may use the replica of these for compiling your files while evaluating your code) **N/A**
4. Driver Details: Does it take care of the options specified earlier(yes/no): **No**
5. Execution status (describe in maximum 2 lines): **The program generates the world and displays correctly, however, the BFS search algorithm doesn’t result the correct path some times**
6. Output Details
   1. Copy and paste the output of four graphs G1-G4 here

G1

G2

G3

G4

Write some more details here for the above graphs, if needed

* 1. Write the following values computed by you (refer the details of R1-R11 in the assignment document). Use appropriate units for the values

R1: R2: R3: R4:

R5: R6: R7: R8:

R9: R10: R11:

1. Declaration: I, Naveen Venkat declare that I have put my genuine efforts in creating the python code for the given programming assignment and have submitted only the code developed by me. I have not copied any piece of code from any source. If the code is found plagiarized in any form or degree, I understand that a disciplinary action as per the institute rules will be taken against me and I will accept the penalty as decided by the department of Computer Science and Information Systems, BITS, Pilani.

**ID** 2015A7PS0078P **Name:** Naveen Venkat

Date: 14/9/17

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Should not exceed three pages