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

Batch No. :

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS**

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

**I Semester 2019-20**

**Programming Assignment-1**

**Coding Details**

**(September 10, 2019)**

*Instruction: Type the details precisely and neatly*

1. ID 2017A7PS0090P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name SAURAV VIRMANI\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Mention the names of Submitted files :
   1. SauravVirmaniCodingDetails.doc
   2. IntelligentVacuumCleaner.py
   3. <filename.ext>
   4. <filename.ext>
   5. <filename.ext>
   6. <filename.ext>
   7. <filename.ext>
2. Total number of submitted files: \_\_\_\_2\_\_\_\_\_\_\_
3. Name of the folder :\_\_\_\_2017A7PS0090P\_\_\_\_\_\_\_\_
4. Have you checked that all the files you are submitting have your name in the top?(yes/no) 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/no) YES
6. Problem formulation
   1. State representation: State representation for the solution is a tuple consisting of two integers.

* The first integer stores the current position of the vacuum cleaner ranging from 0 to row\*columns -1
* The second integer represents the state of the dirt grid, i.e. the dirt representation of a grid in form of a 100 bit integer number.
  1. How is the Initial state generated?

The initial state is generated using the random function to set the position of vacuum cleaner between 0 to rows\*cols-1. The representation of the dirt grid is added as the second member of the tuple.

Then the root node is created out of this with parent action = None, parent = None and cost to reach equal to 0.

* 1. What is the goal state?

Goal state is reaching any of the corner i.e. position in [0,rows-1,rows\*cols-rows, rows\*cols-1] along with the grid dirt integer = 0.

* 1. Are there more than one goal states? Yes there are 4 goal states each represented as a 2 integer tuple.
  2. If yes, then describe all the goal states ?
     1. (0,0)
     2. (rows, 0)
     3. (rows\*(cols-1), 0)
     4. (rows\*cols-1, 0)
  3. State representation in Python (name the construct and give one small example of a state)
     1. Tuple is used for storing the state at a particular node of time.
     2. For example: if the vacuum cleaner is at row =3, col =3,with number of rows =5 and columns=5 and only one dirt position is there at row = 2 and column =0 then state[0]= col\*rows + row = 18 and state[1]=2^(col\*rows+row)= 8 =0x0….0100

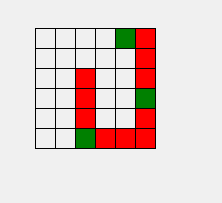
1. Successor function description: Successor function takes as input the current node and then generates a new node with the corresponding parent action, cost to reach this node, generated state of new node and parent of the node.

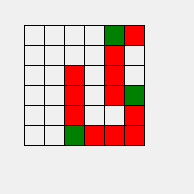
If it is a boundary condition then it returns the same state.

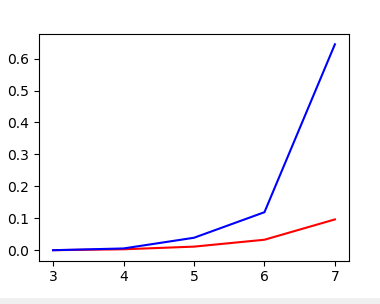
1. BFS (T1) details
   1. Is the search applied on tiles or on states? States.
   2. Error handling and reporting (yes/No): Yes
   3. List the errors handled: KeyError, WrongActionError
   4. Data Structure description for the tree node (in maximum two lines): Object of class type Node with a parent, cost, parent\_action, and state.
   5. Code status (implemented fully/ partially/ not done) : Fully implemented
   6. Maximum depth reached before the failed memory allocation, if happened any? : Not happened till 10\*10 array with 10% dirt. Wasn’t able to analyse.
   7. Maximum room size you are able to handle to reach the goal state within available memory and reasonable time: 10\*10, with percentage of dirt =10
   8. Other limitations of the technique: The algorithm consumes a lot of memory and has exponential space complexity.
2. IDS (T2) details:
   1. Is the search applied on tiles or on states? States
   2. Error handling and reporting (yes/No): Yes
   3. List the errors handled: KeyError, WrongActionError
   4. Data Structure description for the tree node (in maximum two lines): Object of class type Node with a parent, cost, parent\_action, and state.
   5. Code status (implemented fully/ partially/ not done): Fully implemented
   6. Maximum depth reached before the failed memory allocation, if happened any? No, only it was taking a huge amount of time for larger room size and dirt percentage.
   7. Maximum room size you are able to handle to reach the goal state within available memory and reasonable time: 10\*10.
   8. Other limitations of the technique: Consumes a lot of time since it generates a particular state again and again.
3. GUI details
   1. Created the GUI ?(yes/ N0): Yes
   2. Have you created it according to the specifications?(yes/No) Yes
   3. Which module of Python is used for creating graphics? Tkinter
   4. Is this under the standard Python library or not? Yes
   5. If not, why?
   6. Are the window panes working independently? Yes
4. Graphics details:
   1. Is turtle/PyQT graphics working fine for movement of the intelligent vacuum cleaner? Yes
   2. How are you creating the room tiles? By creating rectangles to represent each tile.
   3. How are you showing the dirt? BROWN rectangles for dirt
   4. How are you showing the resting position of the vacuum cleaner? BLACK rectangle for vacuum cleaner.
   5. Are you showing the movement of the vacuum cleaner (turtle cursor) as the execution of T1 goes on? Why or why not? No, because of I only care about the final path.
   6. Are you showing the movement of the vacuum cleaner (turtle cursor) as the execution of T2 goes on? Why or why not? No, because I only care about the final path.
   7. Which functions of Matplotlib are you using? Plot(), add\_subplot()
   8. Are you using any other library such as NUMPY other than the standard Python, PyQT5 and Matplotlib? No.
   9. Any other details:
5. Compilation Details:
   1. Code Compiles (Yes/ No):\_YES\_\_\_\_\_\_\_\_\_\_\_\_\_
   2. Mention the .py files that do not compile:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   3. Any specific function that does not compile:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   4. Ensured the compatibility of your code with the specified Python version(yes/no)\_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)

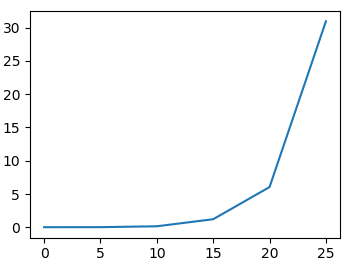
python IntelligentVacuumCleaner.py

1. Driver Details: Does it take care of the options specified earlier(yes/no):\_\_\_\_YES\_\_\_\_\_\_\_
2. Execution status (describe in maximum 2 lines): The code is completely executing, with a completely functional GUI, which helps in selecting the options also.
3. 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: 1041 R2: 56 R3: 119 R4: 29

R5: 0.046 R6: 32308 R7: 56 R8: -

R9: 29 R10: 0.31 R11: - R12: 23.2

1. Declaration: I, \_\_\_\_\_saurav virmani\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (name) 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\_\_\_\_2017A7PS0090P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name:\_\_\_\_\_SAURAV VIRMANI \_\_

Date: \_\_\_10TH SEP 2019\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Should not exceed FOUR pages