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

**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 2017A7PS0117P

Name Satvik Golechha

1. Mention the names of Submitted files :
   1. <cleaner.py>
2. Total number of submitted files: 1
3. Name of the folder :2017A7PS0117P
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: Each state is a class object, which stores as object variables: location of vacuum cleaner, positions of dirt, cost, and path.
   2. How is the Initial state generated? The initial state is generated by calling the class initializer with dirt\_generator(0) and initial location of VC, with cost = 0
   3. What is the goal state?: The goal is reached when entire dirt is cleaned.
   4. Are there more than one goal states?: Yes, there are multiple goal states.
   5. If yes, then describe all the goal states: All locations, all costs are allowed to be a part of a goal state, as long as the dirt gets cleaned.
   6. State representation in Python (name the construct and give one small example of a state)

Class Object. For example, a random state will have these variables stored:

1. self.room = [ [1,0,1], [0,0,1], [1,1,1] ] for a 3x3 room

2. self.location = (1,2), a 2-tuple for the location of VC

3. self.cost = 32, the cost of path till that state is reached

4. self.path = MU -> MR -> S, the action path leading to that state

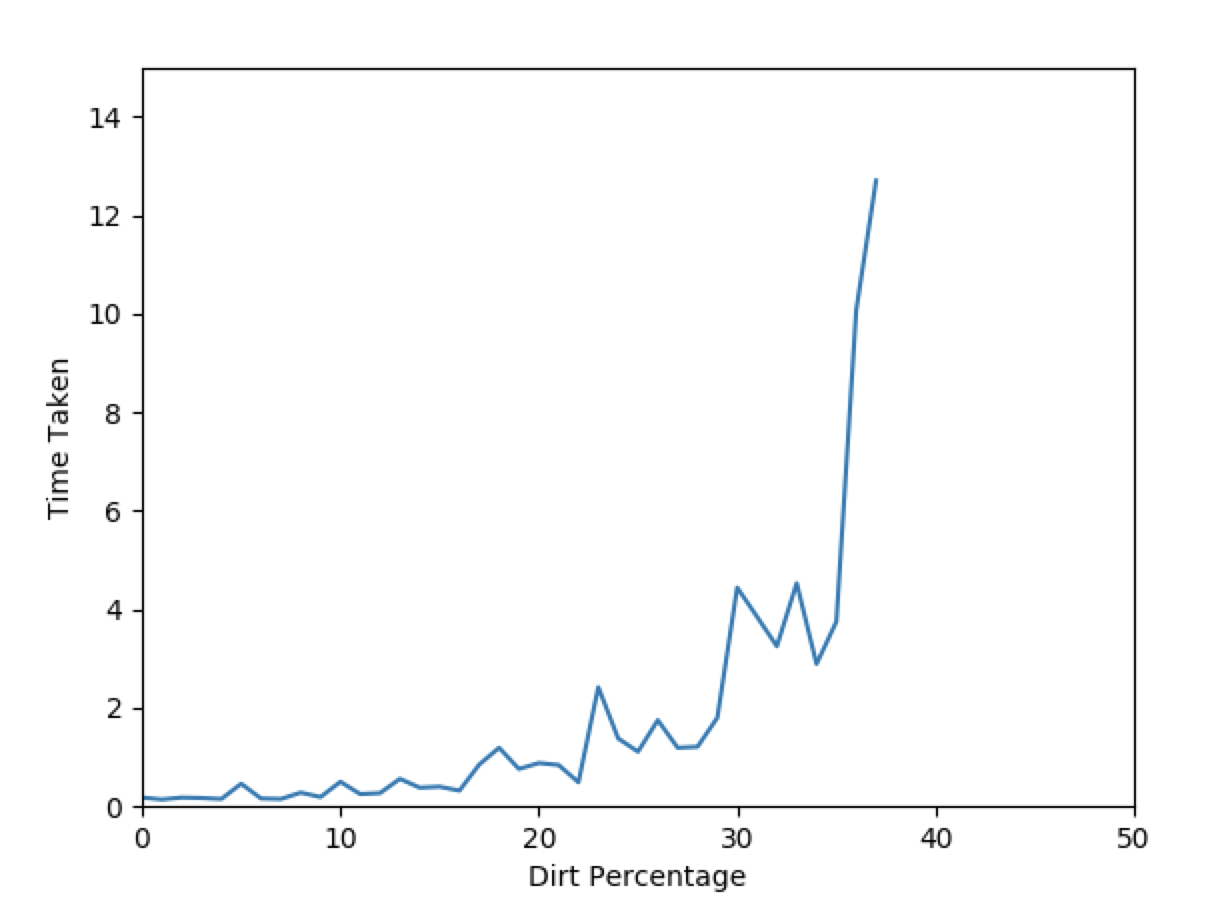
1. Successor function description

The successor function “next\_state(state,action) takes two arguments: a state object, and an action (a string like ‘MU’), and returns a newly created state object, the result of applying the action on that state.

1. BFS (T1) details
   1. Is the search applied on tiles or on states? On States
   2. Error handling and reporting (yes/No): No
   3. List the errors handled:na
   4. Data Structure description for the tree node (in maximum two lines): The tree node is an object of the class ‘state’, and it is a tree with branching factor b=5 for 5 actions and depth d.
   5. Code status (implemented fully/ partially/ not done) Implemented Fully
   6. Maximum depth reached before the failed memory allocation, if happened any? Failed memory allocation did not happen.
   7. Maximum room size you are able to handle to reach the goal state within available memory and reasonable time: Room size 6x6 with 10 percent dirt.
   8. Other limitations of the technique; It takes exponential amount of time with respect to the branching factor ‘b’ and depth ‘d’.
2. IDS (T2) details:
   1. Is the search applied on tiles or on states? On States
   2. Error handling and reporting (yes/No): No
   3. List the errors handled:na
   4. Data Structure description for the tree node (in maximum two lines): The tree node is an object of the class ‘state’, and it is a tree with branching factor b=5 for 5 actions and depth d.
   5. Code status (implemented fully/ partially/ not done) Implemented Fully
   6. Maximum depth reached before the failed memory allocation, if happened any? Failed memory allocation did not happen.
   7. Maximum room size you are able to handle to reach the goal state within available memory and reasonable time: Room size 6x6 with 10 percent dirt.
   8. Other limitations of the technique; It takes exponential amount of time with respect to the branching factor ‘b’ and depth ‘d’.
3. GUI details
   1. Created the GUI ?(yes/ N0): Partially (only graph G4)
   2. Have you created it according to the specifications?(yes/No) Yes
   3. Which module of Python is used for creating graphics? Matplotlib
   4. Is this under the standard Python library or not? Yes
   5. If not, why? na
   6. Are the window panes working independently? na
4. Graphics details:
   1. Is turtle/PyQT graphics working fine for movement of the intelligent vacuum cleaner? na
   2. How are you creating the room tiles? na
   3. How are you showing the dirt? na
   4. How are you showing the resting position of the vacuum cleaner? na
   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 I found it very difficult to implement.
   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 found it very difficult to implement.
   7. Which functions of Matplotlib are you using? plt.plot, plt.ylabel, plt.xlabel, plt.axis, and plt.show, where plt=matplotlib.pyplot
   8. Are you using any other library such as NUMPY other than the standard Python, PyQT5 and Matplotlib? No
   9. Any other details: na
5. Compilation Details:
   1. Code Compiles (Yes/ No):yes
   2. Mention the .py files that do not compile:na
   3. Any specific function that does not compile:na
   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)
6. Driver Details: Does it take care of the options specified earlier(yes/no):Yes
7. Execution status (describe in maximum 2 lines)

The main method acts as the driver and asks for option as input. Hence, we only need to run “python3 cleaner.py” and specify an option as stated from 1-4.

1. Output Details
   1. Copy and paste the output of four graphs G1-G4 here



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: 1616

R2: 609 bytes

R3: 756

R4: 20

R5: 1.09 seconds

R6: 3164

R7: 657 bytes

R8: 165

R9: 44

R10: 1.2 seconds

R11: BFS >> IDS

R12: 22 for BFS and 38 for IDS

1. Declaration: I, Satvik Golechha (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 2017A7PS0117P Name:Satvik Golechha

Date: 10 september 2019

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