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

Batch No. : 2016-17

**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 \_\_\_\_\_\_ 2016A7PS0150P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_Patel Parth\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Mention the names of Submitted files :
   1. Assignment\_1\_Code.py
   2. coding details (PA1).docx
   3. My\_Write\_Up.docx
2. Total number of submitted files: \_\_\_3\_\_\_\_\_
3. Name of the folder :\_\_\_\_\_\_\_2016A7PS0150P\_\_\_\_\_\_\_\_\_
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: The state representation consists of following attributes:
7. Two integers x and y representing the current position of the vacuum cleaner
8. A bytearray of size N\*N (=100), where each byte is 0 (corresponding tile is clean) or 1(corresponding tile is dirty)
9. An integer R denoting the number of tiles currently dirty in the room (This helps to quickly check for goal test instead of we having to iterate 100 elements of bytearray each time to check for goal test).
   1. How is the Initial state generated? Initial x and y coordinates are sampled randomly from 0 to N-1. Similarly, P (input parameter) percentage tiles are randomly chosen to be dirty out of N\*N (=100) tiles.
   2. What is the goal state? (x, y) can be any one of (0,0), (0,N-1), (N-1,0) or (N-1,N-1). All tiles must be clean i.e. all elements of the bytearray must be zero / the integer R should be zero.
   3. Are there more than one goal states? Yes
   4. If yes, then describe all the goal states - Described above
   5. State representation in Python (name the construct and give one small example of a state):

class State:

def \_\_init\_\_(self, N, P):

self.x = random.randrange(0, N, 1)

self.y = random.randrange(0, N, 1)

self.remaining\_tiles = P\*N\*N//100

dirty\_tiles = random.sample(range(N\*N), remaining\_tiles)

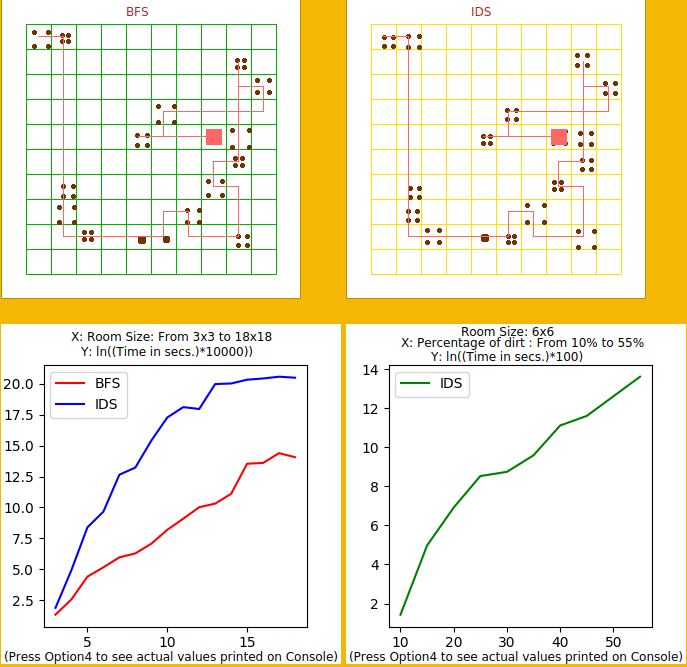
self.floor\_state = bytearray(N\*N)

for tile in dirty\_tiles:

self.floor\_state[tile] = 1

Eg: s = State(10, 10)

1. Successor function description - It takes as input the current state and action to be applied and returns a new Python object representing the next state. If the particular action is not possible from the current state, no new Python object is created and the current state is returned.
2. BFS (T1) details
   1. Is the search applied on tiles or on states? Search is applied on states.
   2. Error handling and reporting (yes/No):
   3. List the errors handled: For efficient memory utilization, collect() method of gc module is called at periodic intervals while performing BFS search to force the Garbage Collector to release unreferenced memory. I also used resource module to set maximum memory available to the program (program generates MemoryError exception on encountering a memory limit), but later found that the resource module works only on certain Unix systems. So, this part of the code is disabled.
   4. Data Structure description for the tree node (in maximum two lines): Node has 3 attributes - Current State, Cost to reach current state from initial state and order of actions taken (as a bytearray) to reach current state from initial state.
   5. Code status (implemented fully/ partially/ not done) - implemented fully
   6. Maximum depth reached before the failed memory allocation, if happened any? 48
   7. Maximum room size you are able to handle to reach the goal state within available memory and reasonable time: 16
   8. Other limitations of the technique:
3. IDS (T2) details:
   1. Is the search applied on tiles or on states? Search is applied on states.
   2. Error handling and reporting (yes/No):
   3. List the errors handled: For efficient memory utilization, collect() method of gc module is called at periodic intervals while performing IDS search to force the Garbage Collector to release unreferenced memory.
   4. Data Structure description for the tree node (in maximum two lines): Same as for BFS.
   5. Code status (implemented fully/ partially/ not done) - implemented fully
   6. Maximum depth reached before the failed memory allocation, if happened any? - N/A as its IDS and not BFS.
   7. Maximum room size you are able to handle to reach the goal state within available memory and reasonable time: 10
   8. Other limitations of the technique: Takes more than reasonable time for room size 10 if depth of tree (i.e. number of movements - right, left, up, down) exceeds 20.
4. 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? PyQt5
   4. Is this under the standard Python library or not? Yes
   5. If not, why? N/A
   6. Are the window panes working independently? N/A
5. 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? Room tiles are created by adding horizontal and vertical lines at equally spaced intervals to the QGraphicsScene object present in PyQt5.
   3. How are you showing the dirt? Dirt is shown as small ellipses (filled with brown color) added at appropriate coordinates to the QGraphicsScene object present in PyQt5.
   4. How are you showing the resting position of the vacuum cleaner? Initial resting position is shown as a red rectangle and the path (red lines) from initial resting position leads to the final resting position.
   5. Are you showing the movement of the vacuum cleaner (turtle cursor) as the execution of T1 goes on? Why or why not? No, movement is shown after T1 is executed and it returns the optimal 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, movement is shown after T2 is executed and it returns the optimal path.
   7. Which functions of Matplotlib are you using? plot() function of Matplotlib.plot() is the main function used for plotting.
   8. Are you using any other library such as NUMPY other than the standard Python, PyQT5 and Matplotlib? No
   9. Any other details: Repeatedly pressing Options 1, 2, 3, and 4 shows the results obtained for one of the 10 randomly generated initial states. I have also provided the functionality to select N (Room Size) and P (Percent of Dirt) and then run BFS/IDS.
6. 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/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) - On the terminal, type the command “python main.py”
7. Driver Details: Does it take care of the options specified earlier(yes/no):\_\_\_Yes\_\_\_
8. Execution status (describe in maximum 2 lines) - The GUI (i.e. options 1, 2, 3, 4) is working as specified. The functionality to run my BFS/IDS code through the GUI is also working fine
9. 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: 344201 nodes R2: 328 bytes R3: 47962 R4: 47 units

R5: 2.55 secs. R6: 2733509 nodes R7: 328 bytes R8: 39

R9: 47 units R10: 28.64 secs. R11: 1229.80 R12: 57.44 units

1. Declaration: I, \_\_\_\_\_\_\_Patel Parth\_\_\_\_\_\_ 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\_\_\_\_2016A7PS0150P\_\_\_\_\_\_\_ Name:\_\_\_\_Patel Parth\_\_\_\_\_\_

Date: \_\_\_\_\_\_10/09/2019\_\_\_\_\_\_\_

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