

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

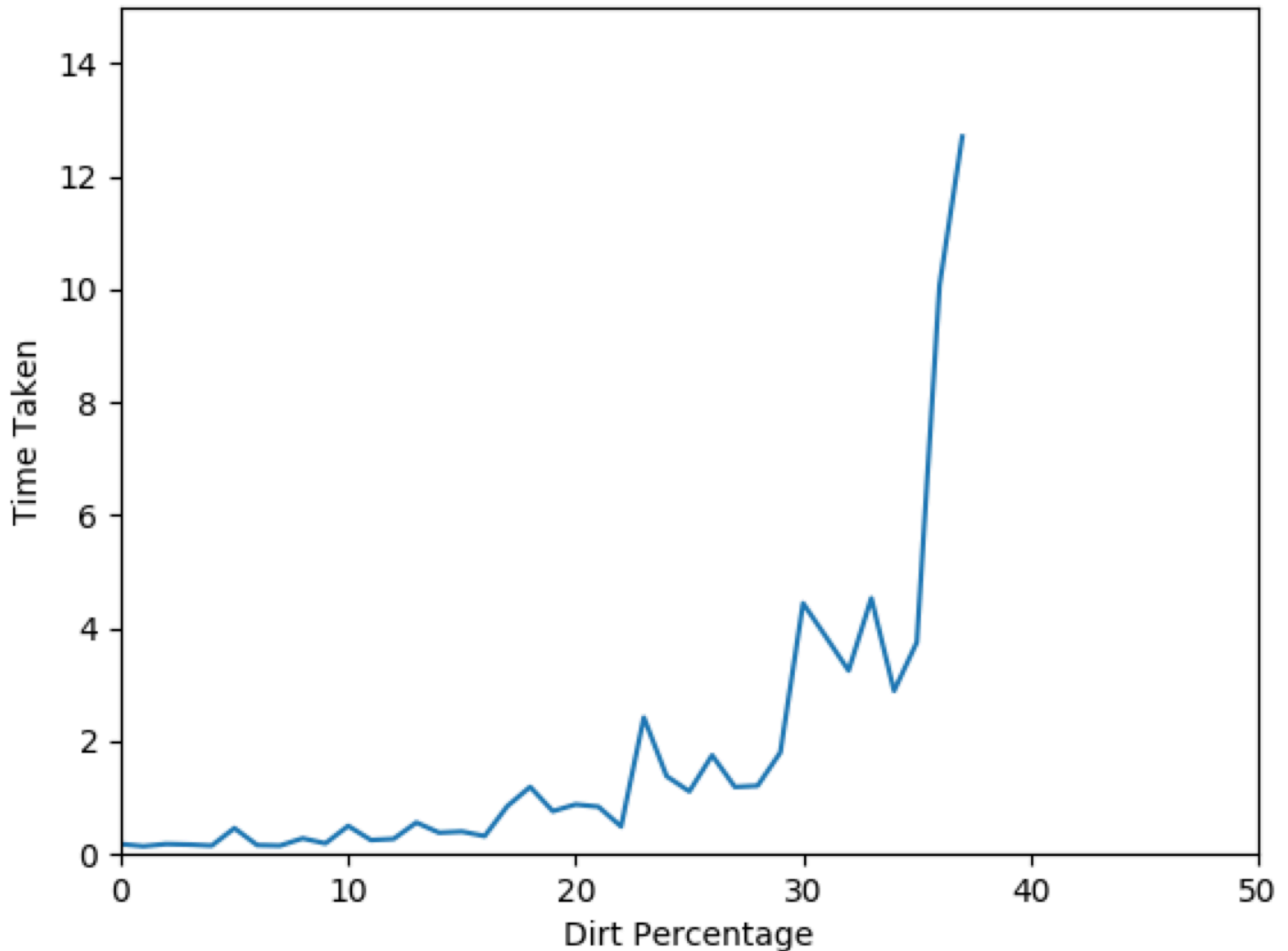
- ID 2017A7PS0117P
Name Satvik Golechha
- Mention the names of Submitted files :
 - <cleaner.py>
- Total number of submitted files: 1
- Name of the folder : 2017A7PS0117P
- Have you checked that all the files you are submitting have your name in the top? YES
- Have you checked that all the files you are submitting are in the folder as specified in 4 (and no subfolder exists)? YES
- Problem formulation
 - State representation: Each state is a class object, which stores as object variables: location of vacuum cleaner, positions of dirt, cost, and path.
 - 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
 - What is the goal state?: The goal is reached when entire dirt is cleaned.
 - Are there more than one goal states?: Yes, there are multiple goal states.
 - 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.
 - 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
- 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.
- BFS (T1) details
 - Is the search applied on tiles or on states? On States
 - Error handling and reporting (yes/No): No

- List the errors handled: N/A
 - 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 .
 - Code status (implemented fully/ partially/ not done) Implemented Fully
 - Maximum depth reached before the failed memory allocation, if happened any? Failed memory allocation did not happen.
 - 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.
 - Other limitations of the technique; It takes exponential amount of time with respect to the branching factor 'b' and depth 'd'.
- IDS (T2) details:
 - Is the search applied on tiles or on states? On the states.
 - Error handling and reporting (yes/No): No
 - List the errors handled:N/A
 - 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 .
 - Code status (implemented fully/ partially/ not done) Implemented Fully
 - Maximum depth reached before the failed memory allocation, if happened any? Failed memory allocation did not happen.
 - 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.
 - Other limitations of the technique: It takes exponential amount of time with respect to the branching factor 'b' and depth 'd'.
 - GUI details
 - Created the GUI?(yes/ N0): Partially (the graphs but not the state and movement)
 - Have you created it according to the specifications? YES
 - Which module of Python is used for creating graphics? Matplotlib
 - Is this under the standard Python library or not? Yes
 - If not, why? N/A
 - Are the window panes working independently? N/A
 - Graphics details:
 - Is turtle/PyQT graphics working fine for movement of the intelligent vacuum cleaner? Not attempted the moving and state viewing GUI.
 - How are you creating the room tiles? N/A
 - How are you showing the dirt? N/A

- How are you showing the resting position of the vacuum cleaner? N/A
- 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 difficult to code.
- 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 difficult to code.
- Which functions of Matplotlib are you using? plt.plot, plt.ylabel, plt.xlabel, plt.axis, and plt.show, where plt=matplotlib.pyplot
- Are you using any other library such as NUMPY other than the standard Python, PyQt5 and Matplotlib? NO
- Any other details: N/A
- Compilation Details:
 - Code Compiles (Yes/ No): Yes
 - Mention the .py files that do not compile: N/A
 - Any specific function that does not compile: N/A
 - Ensured the compatibility of your code with the specified Python version(yes/no) Yes
 - 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 because it does not require multi file compilation scheme.
- Driver Details: Does it take care of the options specified earlier(yes/no): YES,
- 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.
- Output Details
 - 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

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

(these values are for a 4x4 board with 23% dirt and will change a bit each time the code is run)

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

- Declaration: I, Satvik Golechha 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