**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)**

*Instruction: Type the details precisely and neatly*

1. ID 2015A7PS0102P

Name K S Sanjay Srivastav

1. Mention the names of Submitted files :
   1. dirtGenerator.py
   2. GUI.py
   3. T1.py
   4. T2\_1.py
   5. T2\_2.py
   6. main.py
   7. 2015A7PS0102P.docx
2. Total number of submitted files: 7
3. Name of the folder : 2015A7PS0102P
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: A State is a list having the x-position, y-position of the tile we are on and also the depth from the root node i.e., a resting position in case of uninformed search and h2 heuristic , and for h1 heuristic I modified my state to contain a list of all the parent states it has visited till root state.
   2. How is the Initial state generated? My initial state is always [0,0, max\_depth] where max\_depth is generated by the caller function which initially is 2\*p-1 where p is the number of dirt tiles given.
   3. What is the goal state? A state where all the tiles are cleaned for the search technique and for turtle a state which is resting position.
   4. Are there more than one goal states? Yes
   5. If yes, then describe all the goal states: Primary goal State is to find if all the tiles are clean and secondary goal state is to find the nearest resting position to go there in case of uninformed Search whereas in case of informed search there are three goal states as my technique uses best-first approach ,one goal test is to find the first best dirt, and another global goal test to find if all dirts are cleared and tertiary goal test if resting point reached.
   6. State representation in Python (name the construct and give one small example of a state)

State=[x-pos, y-pos, depth ] Example: [0,0,0]

1. Successor function description: Return a list of all possible children for a given state eliminating nodes already in the path to reach this state thus reducing the branching factor to at most 3.Also sucking operation is not a part of my children as all the cleaning is done once the path is generated.
2. Uninformed Search Technique (T1) details
   1. Technique used for search: Iterative Deepening Search.
   2. Reason for selecting this technique over the other two: Because this gives the more accurate answer.
   3. Is the search applied on tiles or on states? Only state as the goal test here checks on entire grid.
   4. Error handling and reporting (yes/No): yes
   5. List the errors handled: Boundary conditions are handled as the turtle cannot cross the grid and also index out of range exceptions are eliminated.
   6. Data Structure description for the tree node (in maximum two lines): Tree node has a state and if a node is to be added to the tree all the nodes that are at more depths in our explored path are eliminated thus adding each branch at a time to a parent.
   7. Code status (implemented fully/ partially/ not done) fully, but it is really slow for larger grid sizes.
3. Informed search Technique (T2) details:
   1. Technique used for search: Best-First greedy approach using IDFS and BFS for each heuristic respectively.
   2. Reason for selecting this technique over others: This way the total memory at any time is proportional to
   3. Does this technique look at a tile? Yes, if it is a dirt it cleans it and calls the search technique from there.
   4. Does this technique use a state? Yes, as heuristic decide the children.
   5. Code status (implemented fully/ partially/ not done) Fully.
   6. Define the heuristics (in words) used in your program
      1. h1 : in my successor function, my priorities go from left ,top, right to bottom
      2. h2 : reverse priorities from h1
   7. Compute the heuristic values for the following windows with dirt in the neighborhood of the centre position of the vacuum cleaner. If l denotes left, r-right, t-top and b-bottom each heuristic gives the ordering of successors
      1. h1 = rltb , h2 = rbtl
      2. h1 = tlrb , h2 = brtl
      3. h1 = tlrb , h2 = rbtl
      4. h1 = ltrb , h2 = tbrl

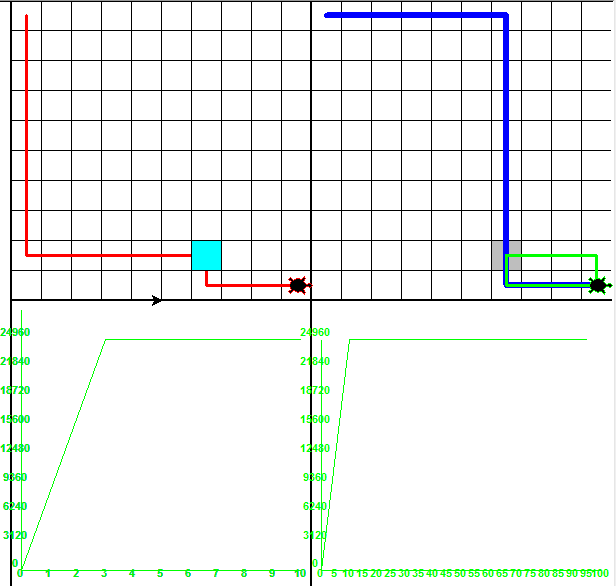
(i)

(ii)

(iii)

(iv)

1. GUI details
   1. Created the GUI (yes/ N0):yes
   2. Have created it according to the specifications?(yes/No)Yes
   3. Which module of Python used for creating graphics? Turtle
   4. Is this under the standard Python library or not? Yes
   5. If not, why?
   6. Are the window panes working independently? No
2. Graphics details:
   1. Is turtle graphics working fine for movement of the intelligent vacuum cleaner? Yes
   2. How are you creating the room tiles? Using 10 horizontal and 10 vertical lines.
   3. How are you showing the dirt? Coloring those tiles with grey.
   4. How are you showing resting position of the vacuum cleaner? Turtle’s final position is resting position
   5. Are you showing the movement of the vacuum cleaner (turtle cursor) as the execution of T1 goes on? Why? No. It only moves once the entire path gets generated.
   6. Are you showing the movement of the vacuum cleaner (turtle cursor) as the execution of T2 goes on? Why? Yes, because I have used best-first approach and hence it moves to the best one and searches again and hence path is generated with the turtle’s movement.
3. Compilation Details:
   1. Code Compiles (Yes/ No):Yes
   2. Mention the .py files that do not compile: None
   3. Any specific function that does not compile: None
   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 main.py”
4. Driver Details: Does it take care of the options specified earlier(yes/no):Yes
5. Execution status (describe in maximum 2 lines) All the codes are properly executing but uninformed search is taking more time to complete as the memory usage is very high but it gives the optimal path. Informed search (best-first ones) are not optimal but the results are almost instantaneously generated.
6. Output Details
   1. Copy and paste the output of four graphs G1-G4 here



Write some more details here for the above graphs– For G3,G4 y-axis indicates time in milli seconds.

* 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: 2846723 nodes R2: 88 bytes R3: 21 nodes in stack R4: 37 units

R5: 17.087 sec R6: 491531 nodes (h1),46 nodes(h2) R7: (h1)88 bytes,(h2)80bytes

R8: (h1)37 units, (h2)17 units R9: (h1)33.684 sec, (h2)3.5 sec R10: (T1)4224KB, (h1)25406KB, (h2) 0.0KB R11: (T1) average 31 units, (T2(h2)) average 17 units

1. Declaration: I, K S Sanjay Srivastav, 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: 2015A7PS0102P Name: K S Sanjay Srivastav

Date: 14-9-2017

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