**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 \_\_2014A7PS0095P\_\_\_

Name \_\_Lakshit Bhutani\_\_\_

1. Mention the names of Submitted files :
   1. analysis.py
   2. dirtgenerator.py
   3. driver.py
   4. search.py
   5. utils.py
2. Total number of submitted files: \_\_\_\_\_5\_\_\_\_\_
3. Name of the folder :\_\_\_2014A7PS0095P\_\_\_
4. Have you checked that all the files you are submitting have your name in the top?(**yes**/no)
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)
6. Problem formulation
   1. State representation: tuple of current position, path followed uptil now and the number of dirt cells cleaned. Informed search has heuristic value as well.
   2. How is the Initial state generated? Initial state is returned by GetStartState() method which takes the initial position and returns the required state tuple
   3. What is the goal state? The goal state is when all the dirt is cleaned and the vacuum cleaner is on one of the resting positions
   4. Are there more than one goal states? Yes
   5. If yes, then describe all the goal states : After cleaning the entire dirt, cleaner can rest at any of the 4 resting positions i.e. corners
   6. State representation in Python (name the construct and give one small example of a state): State is a python tuple. Example of state for uninformed search with initial position as left top most corner having no dirt – [(0,0), [(0,0)], 0]
7. Successor function description : Successor function checks all the neighbours for the current location and makes it the current position, appends it to the path traversed, updates the number of tile cleaned and inserts the appropriate heuristic value (informed)
8. Uninformed Search Technique (T1) details
   1. Technique used for search: Depth First Search (DFS)

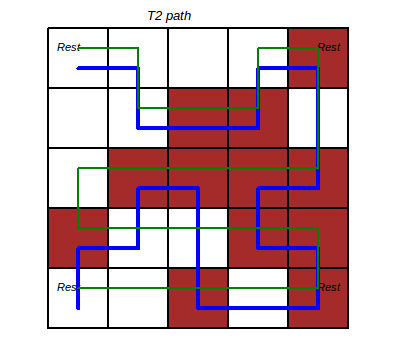
* 1. Reason for selecting this technique over the other two : DFS takes lesser memory and runs faster on average

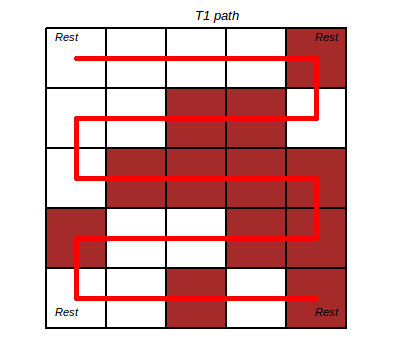
* 1. Is the search applied on tiles or on states? States in search tree
  2. Error handling and reporting (**Yes**/No):
  3. List the errors handled: Constraint check on dimension, dirt percentage and error reporting of any search method fails to find a path
  4. Data Structure description for the tree node (in maximum two lines): Tree node is a tuple representing the state having currnt position, path traversed, tiles cleaned and heuristic value
  5. Code status (**implemented fully**/ partially/ not done)

1. Informed search Technique (T2) details:
   1. Technique used for search: Greedy best first search
   2. Reason for selecting this technique over others: It is low time coimplexity
   3. Does this technique look at a tile? Yes (neighborhood)
   4. Does this technique use a state? Yes
   5. Code status (implemented fully/ **partially**/ not done) : Works for small dimension
   6. Define the heuristics (in words) used in your program
      1. h1 : sum of x and y coordinates of current position added with number of neighbouring dirty cells and added 2 points if cell is itself dirty
      2. h2 : number of neighbouring dirty cells and added 2 points if cell is itself dirty
   7. Compute the heuristic values for the following windows with dirt in the neighborhood of the center position of the vacuum cleaner.
      1. h1 = 5 , h2 = 3
      2. h1 = 8 , h2 = 6
      3. h1 = 6 , h2 = 4
      4. h1 = 5 , h2 = 3
2. GUI details
   1. Created the GUI (**yes**/ N0):
   2. Have created it according to the specifications?(**yes**/No)
   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? NA
   6. Are the window panes working independently? They work sequentially
3. Graphics details:
   1. Is turtle graphics working fine for movement of the intelligent vacuum cleaner? Yes
   2. How are you creating the room tiles? Grid of n by n
   3. How are you showing the dirt? Colouring the dirty square brown
   4. How are you showing the resting position of the vacuum cleaner? Rest squares are marked with the word “Rest”
   5. Are you showing the movement of the vacuum cleaner (turtle cursor) as the execution of T1 goes on? Why? Yes, to trace the path followed
   6. Are you showing the movement of the vacuum cleaner (turtle cursor) as the execution of T2 goes on? Why? Yes, to trace the path followed
4. 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 driver.py\_\_*
5. Driver Details: Does it take care of the options specified earlier(**yes**/no):\_\_\_\_Yes\_\_\_\_\_
6. Execution status (describe in maximum 2 lines)

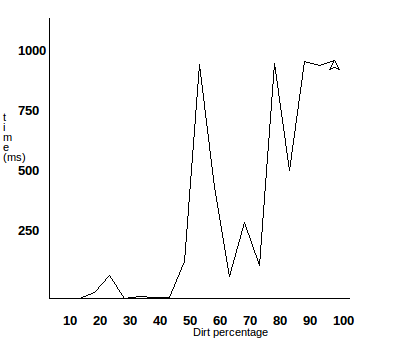
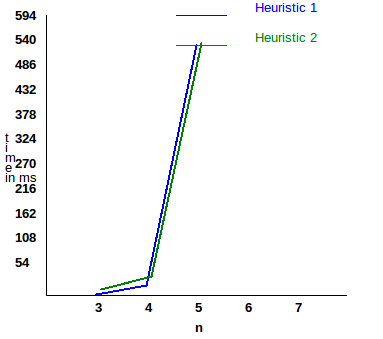
Uninformed search works for all room dimensions whereas Informed search works for room with N = 5 reasonable fast, N=6 slowly and very slow fo N >= 7

1. 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

These are readings for N = 5 and 50% dirt

* 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: 40 nodes R2: 96 bytes R3: 16 R4: 60 units

R5: 0.867 ms R6: 4771 (h1), 2336(h2) R7: 104(h1), 104(h2) R8: 52 units(h1), 56 units(h2) R9: 143.668ms (h1), 39.113ms (h2) R10: 3200 bytes (T1), 312708 bytes (T2) R11: 60 units (T1), 54 units (T2)

1. Declaration: I, \_\_Lakshit Bhutani\_\_\_\_\_ (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 \_\_\_2014A7PS0095P\_\_\_\_ Name:\_\_\_Lakshit Bhutani\_\_\_\_\_\_\_

Date: \_\_\_\_Sept 14, 2017\_\_\_\_\_\_\_\_\_\_

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