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

Batch No. :

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS**

**Artificial Intelligence (BITS F444/ CS F407)**

**I Semester 2018-19**

**Programming Assignment-4**

**Coding Details**

**(November 2, 2018)**

*Instruction: Type the details precisely and neatly*

1. ID \_\_\_\_\_\_\_\_\_\_2017H1030130P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Santosh Desai\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Mention the names of Submitted files :
   1. Api.py
   2. Driver.py
   3. Helper.py
   4. Gui.py
2. Total number of submitted files: \_\_\_\_\_\_4\_\_\_\_\_
3. Name of the folder :\_\_\_\_\_\_\_\_2017H1030130P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
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. List of variables (Specify all variables):

One variable is used, each for one place in the grid. A 4x4 grid will have 16 variables for 16 places.

* 1. Value domains of variables (Also list the variables against each value domain correspondingly)

Initial domains of all the variables is [1, num] where num is the total number of places in the grid.

* 1. Mention the constraints

Sum of each row, sum of each column, sum of both the diagonals are all equal.

1. Data structure used
   1. Constraint graph node structure:

Array of values

* 1. Constraint graph edge structure:

It is a pair <start\_node,end\_node>

* 1. Constraint graph (Adjacency list/ adjacency matrix/ any other(specify)

Adjacency List with hashing

* 1. How are you maintaining value domains as you go with search process?

Hashing the domains (it also helps in forward checking)

1. DFS + backtracking technique details
   1. Variable ordering used (List heuristics used):
      * 1. Default order, 1,2,3….
        2. MRV + Degree heuristic
        3. Siam’s method
   2. Node structure for DFS:

Each node contains [domains, hidden values of domains as a separate list, constraints, values to be assigned, assignments]

* 1. Method for assignment of a value to a variable and backtracking:

Maintaining a dictionary of assignments for easy retraction

* 1. How is edge node of your adjacency list (constraint graph) useful in deciding upon which constraint module( or modules) to use for testing the violation of the constraints while you assign a value to a variable?
     1. Domain reduction : AC3 algorithm support
     2. Checking for inconsistencies
     3. Forward Check
     4. Backtracking
  2. Total number of nodes generated for assignment of values to all variables: 139624 (worst case for 4x4 grid)
  3. Write the statistics here as asked (for a 4x4 grid)

R1 = 139624 R2 = 3184 R3 = 16

R4 = 11.288 R5= 167184

* 1. Code status (implemented fully/ partially/ not done) implemented fully

1. DFS+ Backtracking using constraint propagation:
   1. Explain the method for constraint propagation. How are you updating the value domains? What do you do with the value domains of the variables when you backtrack while performing DFS?

Method : Forward Check and AC3 are supported.

Updation : Reducing possible domains after assignment

Backtrack : Put the values back (via a system stack)

* 1. Total number of nodes generated using the above technique : 90474
  2. Write the statistics here as asked

R6 = 90474 R7 = 0.352017 R8 = 18.6789

1. Code status (implemented fully/ partially/ not done) : implemented fully

1. Comparative analysis

Fill in the following information

|  |  |  |
| --- | --- | --- |
|  | DFS+BT | DFS+BT+Constraint propagation |
| Average number of nodes created | 139624 | 90474 |
| Average time taken | 11.288 seconds | 18.6789 seconds |

1. 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)

$python driver.py

1. Driver Details: Does it take care of the options specified earlier(yes/no):\_\_\_yes\_\_\_\_\_\_\_\_
2. Execution status (describe in maximum 2 lines) : Compiles and runs successfully
3. Declaration: I, \_\_\_\_\_\_\_\_\_\_Santosh Kumar Desai\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (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\_\_\_\_\_\_2017H1030130P\_\_\_\_\_\_\_\_ Name:\_\_Santosh Kumar Desai\_\_\_\_

Date: \_\_\_02/11/18\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Should not exceed four pages