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

Batch No. :

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS**

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

**I Semester 2019-20**

**Programming Assignment-4**

**Coding Details**

**(November 1, 2019)**

*Instruction: Type the details precisely and neatly*

1. ID : 2017A7PS0117P

Name : Satvik Golechha

1. Mention the names of Submitted files :
   1. Image.jpeg
   2. eminent.py
   3. Problem Statement.pdf
2. Total number of submitted files: 3
3. Name of the folder: 2017A7PS0117P
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):

Variables are eminent personalities.

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

Value domains of each variable are all the free hours for each eminent personalities.

* 1. Mention the constraints

For each group, consider all pairs of persons that group wants to meet,

Each pair gives a binary constraint, where the two values must be different.

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

Binary graph: nodes are variables as eminent personalities

* 1. Constraint graph edge structure:

Each edge denotes a constraint

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

Adjacency matrix (both image and implementation using matrixify()

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

Value domains are stored and maintained in the form of lists, and if user wants to use constraint propagation, the domain values are reduced because it wants to maintain arc consistency.

1. DFS + backtracking technique details
   1. Variable ordering used (List heuristics used):
      1. Minimum Remaining Value
      2. Degree constraint.
   2. Node structure for DFS:

Node denotes a variable, in this case eminent personalities

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

Depth first search is used to search the entire tree. Assignment is done through either a heuristic or the first value.

* 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?

Each edge is a constraint which is checked for consistency while going down the tree while increasing the size of partial assignments.

* 1. Total number of nodes generated for assignment of values to all variables:
  2. Write the statistics here as asked

R1 = R2 = R3 = NA

R4 = R5=NA

* 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?

AC-3 is implemented as given in the textbook.

Value and variable domains depend on the heuristic used.

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

R6 = R7 = R8 =

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 |  |  |
| Average time taken |  |  |

1. Compilation Details:
   1. Code Compiles (Yes/ No): YES
   2. Mention the .py files that do not compile: NA
   3. Any specific function that does not compile: NA
   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) YES
2. Driver Details: Does it take care of the options specified earlier(yes/no): YES
3. Execution status (describe in maximum 2 lines) The code compiles and executes successfully,
4. 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: 1 November, 2019

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