**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-3**

**Coding Details**

**(October 26, 2017)**

*Instruction: Type the details precisely and neatly*

1. ID 2015A7PS0111P

Name T NAGA DATTA MADHU KIRAN

1. Mention the names of Submitted files :
   1. combinations.py
   2. constraint.py
   3. main.py
   4. pre\_module.py
   5. com\_gen.py
   6. testcase1csv
   7. static\_TT\_DFS\_BT\_prop\_WITH\_heur.csv (Precomputed\_timetable)
   8. static\_TT\_DFS\_BT\_prop\_WITHOUT\_heur.csv (Precomputed\_timetable)
   9. static\_TT\_DFS\_BT\_WITH\_heur.csv (Precomputed\_timetable)
   10. static\_TT\_DFS\_BT\_WITHOUT\_heur.csv (Precomputed\_timetable)
   11. DFS\_BT\_propagate.csv (Dynamically generated one)
   12. DFS\_BT.csv (Dynamically generated one)
   13. 2015A7PS0111P.docx
   14. constraintGraph.pdf
2. Total number of submitted files: 14
3. Name of the folder :2015A7PS0111P
4. Have you checked that all the files you are submitting have your name in the top? 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
6. Problem formulation
   1. List of variables (Specify all variables):

C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11

(indeed each variable is further divided into further three variable each for lecture,lab,tutorial of that particular course.)

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

basic\_domain={11,12,13,14,16,17,18,

21,22,23,24,26,27,28,

31,32,33,34,36,37,38,

41,42,43,44,46,47,48,

51,52,53,54,56,57,58,

61,62,63,64}

so if a course contain 'l' number of lectures and 't' number of tutorials and 'p' number of labs then

domain of that variable ={[basic\_domain]^l,[basic\_domain]^t,[basic\_domain]^p}

* 1. Constraint graph : Draw the constraint graph neatly on a separate A3 paper. Encircle the variables to represent the nodes and label each edge with appropriate constraints numbered from 1 to 11 as mentioned in the problem specification document. Upload the scanned copy of the above hand drawn constraint graph as file **constraintGraph.pdf**.

Have you uploaded the scanned copy as above? Yes

* 1. Specify the number of nodes in your constraint graph. 12
  2. Specify the number of edges in your constraint graph. 59
  3. How do you use the package information in constructing the constraint graph?

Pseudocode:

For all package:

for any two courses in a package:

there must exist an edge between those two course

(since package can be taken by students.)

* 1. How do you use professor-courses packages in your constraint graph?

Pseudocode:

For all professor-courses packages:

for any two courses in a professor-courses packages:

there must exist an edge between those two course

(since any professor have to take all of his/her courses)

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

name\_variable : list[adjacent\_variables]

* 1. Constraint graph edge structure:

since key and value pair used as node structure edge is direct hashing of key to get corresponding value

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

Adjacency list

1. DFS + backtracking technique details
   1. Variable ordering used:

for heuristic based on degree heuristic ordering is done

for without heuristic ordering is sorting based on name

* 1. Node structure for DFS:

Node contain domain, name, state(dictionary)

state={'course\_name':lecture\_slots,tutorial\_slots,

lab\_slots([1st\_batch,2ndBatch]),assigned\_flag]}

domain={'course\_name':[l\_domain,t\_domain,p\_domain]}

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

Pseudocode:

current variable is assigned to value

and call DFS\_BT on the remaining variable if this results in true then return the corresponding schedule else undo the assignment and add the variable to variable list. If it fails after exhaustive search then return False

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

R1 = R2 = R3 =

R4 = R5=

* 1. Code status 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?

Pseudocode:

current variable is assigned to a value

have a copy of original domain

call propagate function to reduce the domain based on constraint graph

and call DFS\_BT\_propagate on the remaining variable if this results in true then return thecorresponding schedule else undo the assignment, changing the domain to the copy that we have and add the variable to variable list. If it fails after exhaustive search then return False

Pseudocode:

propagate function :it removes the value in the current assignment in the domains of adjacent nodes in the adjacent list(constraint graph)

return the updated domain.

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

R6 = R7 = R8 =

d. Code status -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. Output files
   1. Created the output files DFS\_BT\_propagate.csv and DFS\_BT.csv (yes/ No):yes
   2. Have named the files according to the specifications?(yes/No):yes
   3. Created the file constraintGraph.pdf (yes/no): yes
2. 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 com\_gen.py

>>python main.py

1. Driver Details: Does it take care of the options specified earlier(yes/no):yes
2. Execution status (describe in maximum 2 lines)

Option 1,2,3,4 are executed as per specification with no errors.

1. Declaration: I,T Naga Datta Madhu Kiran 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 2015A7PS0111P Name:T Naga Datta Madhu Kiran

Date: 26-Oct-2017

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