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

**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 2017A7PS0093P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_AYUSH JAIN\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Mention the names of Submitted files :
   1. csp.py
   2. partial\_assignment.txt
   3. Coding details PA4
2. Total number of submitted files: 3
3. Name of the folder :\_\_\_\_\_\_2017A7PS0093P\_ASS4\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Have you checked that all the files you are submitting 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
6. Problem formulation
   1. List of variables (Specify all variables):

variables were basically the nobel laureates: N1, N2, ……………….., NM

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

domains were the time slots in which each nobel laureate is available.

{N1: [2, 5, 7],

N2: […….],

.

..

N20: [......]}

* 1. Mention the constraints

(Ni, Nj: Ni and Nj belongs to the same group Gk, and are assigned the same time slot ti)

1. Data structure used
   1. Constraint graph node structure: Dictionary, with key as node label and values as its domains
   2. Constraint graph edge structure: Dictionary with key as node label and value as the node labels of neighbors
   3. Constraint graph (Adjacency list/ adjacency matrix/ any other(specify): Adjacency dictionary, with node label as key and value being the adjacency list for that node.
   4. How are you maintaining value domains as you go with search process? Instead of modifying the original domains, I send copy of the domains to backtrack, hence if backtrack fails then I can restore the original domain.
2. DFS + backtracking technique details
   1. Variable ordering used (List heuristics used): MRV and Degree Heuristic
   2. Node structure for DFS: A new dictionary called assignment is created. Which keeps storing the partial assignment. Initially it is empty. We go to search through the variable chosen as per variable ordering. We assign it to element of its domain one by one, such that the assignment remains consistent.

* 1. Method for assignment of a value to a variable and backtracking: Value to variable is assigned in order of its original domain. We try finding a value such that the assignment is consistent. If it is successful we go one layer deep and try assignment to the next variable. If the value list is exhausted and we cannot find any valid assignment we backtrack.
  2. 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? I only need to check if any neighbor of the assigned variable, which I get from my adjacency dict, is already there in the assignment and has the same time slot as the value we are trying to assign.
  3. Total number of nodes generated for assignment of values to all variables: 53270
  4. Write the statistics here as asked

R1 = 53270 R2 = 248 bytes R3 = 3968 bytes

R4 = 2.484 sec R5= Degree: 313; MRV: 2312

* 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?

Imagine an empty queue. I add all the edges of the graph involving the assigned variable and its neighbor, excluding those neighbors which are already assigned values. Then I see if that edge is arc consistent. We prune domains by deleting those variables from the domains of the neighbors which cannot be satisfied for any assignment. We add the edge to the neighbors of the variable whose domain is modified, to our queue, provided that neighbor of neighbor is not in assignment.

I only pass copy of the domains and never the original domains to backtrack, so if the backtrack fails I have the original domain with me safe.

* 1. Total number of nodes generated using the above technique: 13658
  2. Write the statistics here as asked

R6 = 13658 R7 = 0.743 R8 = 1.53

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 | 53270 | 13658 |
| Average time taken | 2.484 | 1.53 |

1. 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) python3.7 csp.py

It will prompt you for choosing default values (given in problem) or enter own values. An easy interface is made for user input. Then it asks for whether you want to use AC3 or not; then do you want to use heurestics. Finally the partial assignment are shown after every 100 nodes. Finally on top of the screen, it is displayed whether assignment was successful or not. Further you can see the precomputed statistics on the screen by clicking on “see statistics” button. Finally you can click on continue to start the process again.

1. Driver Details: Does it take care of the options specified earlier(yes/no):\_\_\_\_\_yes\_\_\_\_\_\_
2. Execution status (describe in maximum 2 lines): Complete
3. Declaration: I, \_\_\_\_\_\_\_\_\_\_\_\_AYUSH JAIN\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (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\_\_\_\_\_\_\_\_\_\_\_\_2017A7PS0093P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name:\_\_\_\_\_\_AYUSH JAIN\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_1 NOVEMBER, 2019\_\_\_\_\_\_\_\_\_\_

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