DISCO Project

PROBLEM FORMULATION

Problem Title:

Provide all possible allotments of courses to professors according to course preferences filled by them.

Description:

Within a department, there are "n" faculty members categorised into three distinct groups: "x1," "x2," and "x3". Professors in each category are assigned different course loads, with "x1" handling 0.5 courses per semester, "x2" taking 1 course per semester, and "x3" managing 1.5 courses per semester. There is no prioritisation among faculty members within the same category.

There is an application of graph theory in the backtracking code. In the context of the code, each faculty member can be considered as a node in the graph. The edges between the nodes represent the possibility of assigning a course to a professor. Backtracking is analogous to finding all possible paths in a graph.

Input:

Input File “file_name.txt”.

It consists of n lines (n professors).
Each line begins with the name of the professor followed by the number of courses he can offer. It is then followed by the preferences of “ugcdc”, then by “ugele”, then “hdcdc” and finally “hdele”. Preferences are separated by commas (,). End of the input file is denoted by “END”.

Output:

Output is a text file “Output.txt” which includes all the possible allotments of courses to professors.

Constraints:

1. Every professor should be assigned courses as per his/her course load (not more or less).
2. Every CDC course must be assigned to any of the professors.
3. A course can only be assigned to a faculty member if it is present in their preference list only.
4. No constraints on allotment of electives.

Constraints Explanation:

1. We need to make sure that no CDC is left unassigned. There should be enough number of professors for that in input file.
2. Professor’s course load should be satisfied. For that if after assigning CDC his/her course load is less than filled by him then he/she should be assigned electives as per the preferences. Also total of course loads of professors should be an integer.

Problem Solving Approach and Algorithm

We start by taking the input from a text file and store it in a map whose key is professor's name and value is a vector of 5 vectors. The first of these 5 vectors holds the workload of the professor and the rest hold the different types of courses' preference of the prof. We then make an inverse map of which key is course and value is vector of profs who have filled the course in their preferences.

We then start allotting the CDCs in base case. In base case, we allot all the CDCs that compulsorily need to be allotted in order to generate a valid combination. We then use backtracking to allot the remaining CDCs. In our backtracking approach, we use course (remaining CDCs) as level.

Out of all possible combinations, we first remove the duplicates. We then check whether each particular combination can have a valid final combination after satisfying professors' workload constraints.

We then use backtracking again to allot electives where level is course again. This backtracking is done on all possible combinations of CDC allotments.

We then use a set to store all the combinations to avoid duplicates.

Output is then stored into another .txt file.