Lab Task1: Course Scheduling System

Objective: Develop a Course Scheduling System for a university semester that assigns time slots and classrooms to courses while satisfying a set of constraints. The goal is to ensure that no student has overlapping classes, instructors are assigned to one class at a time, and the room capacity meets the course enrollment requirements.

Description:

The university offers a variety of courses, each with a set number of students enrolled, a list of possible time slots, and room preferences. Instructors can teach multiple courses but cannot be in two places at once. Rooms have different capacities and some are equipped with special equipment needed for certain courses.

Task:

Implement a CSP algorithm to create a schedule that:

- Assigns a time slot and a classroom to each course.
- Ensures no instructor is teaching two courses at the same time.
- Avoids student schedule conflicts.
- Matches room capacity with course enrollment.
- (Optional) Considers room equipment requirements for specialized courses.

Requirements:

1. Variables: Each course to be scheduled is a variable.

2. Domains: For each course, the domain is the combination of all possible time slots and classrooms available.

3. Constraints:

- No two courses taught by the same instructor can have the same time slot.
- No student can be enrolled in two courses that occur at the same time.
- The number of students enrolled in a course cannot exceed the capacity of the assigned classroom.
- (Optional) Certain courses require specific room equipment (e.g., a projector, lab equipment); these requirements must be met.

Visualize your output by using library of your own choice

Task2: Optimal Event Planning and Guest Seating Arrangement

Objective: Develop an algorithm to plan guest seating at an event, such as a wedding or gala dinner, ensuring all guest preferences and social distancing rules are met. The

goal is to arrange guests in a way that maximizes overall guest satisfaction while adhering to any constraints regarding table sizes, guest relationships, and specific requests.

Description

You're tasked with planning the seating arrangement for an event where guests have preferences about whom they would like to sit with or avoid. Some tables are larger than others, and a few guests may require special accommodations, such as easy access to exits or specific dietary restrictions affecting table placement.

Your Task

Implement a CSP algorithm to generate a seating arrangement that:

- Assigns guests to tables according to their seating preferences and any special requirements.
- Ensures no guest is seated at a table with someone they wish to avoid.
- Maximizes the number of preferences satisfied across all guests.
- Adheres to table size limitations and special accommodation needs.

Requirements:

1. Variables: Each guest to be seated at the event.

2. Domains: For each guest, the domain includes all possible tables where they could be seated.

3. Constraints:

- No guest is seated at a table with someone they wish to avoid.
- Each table has a maximum seating capacity that cannot be exceeded.
- Guests with special accommodation needs must be seated at tables that meet those needs (e.g., proximity to exits, wheelchair accessibility).
- (Optional) Some guests may have a strong preference to be seated together; these preferences should be satisfied as much as possible.

Visualize your output by using library of your own choice