Question:

Course Scheduling System: Priya is a member from the training department of a company who has to prepare a training plan that has different courses. Few of these courses need some prerequisite courses to be completed. The goal is to prepare a training plan such that all the courses are included in the correct order and the learning is on track.

Solution:

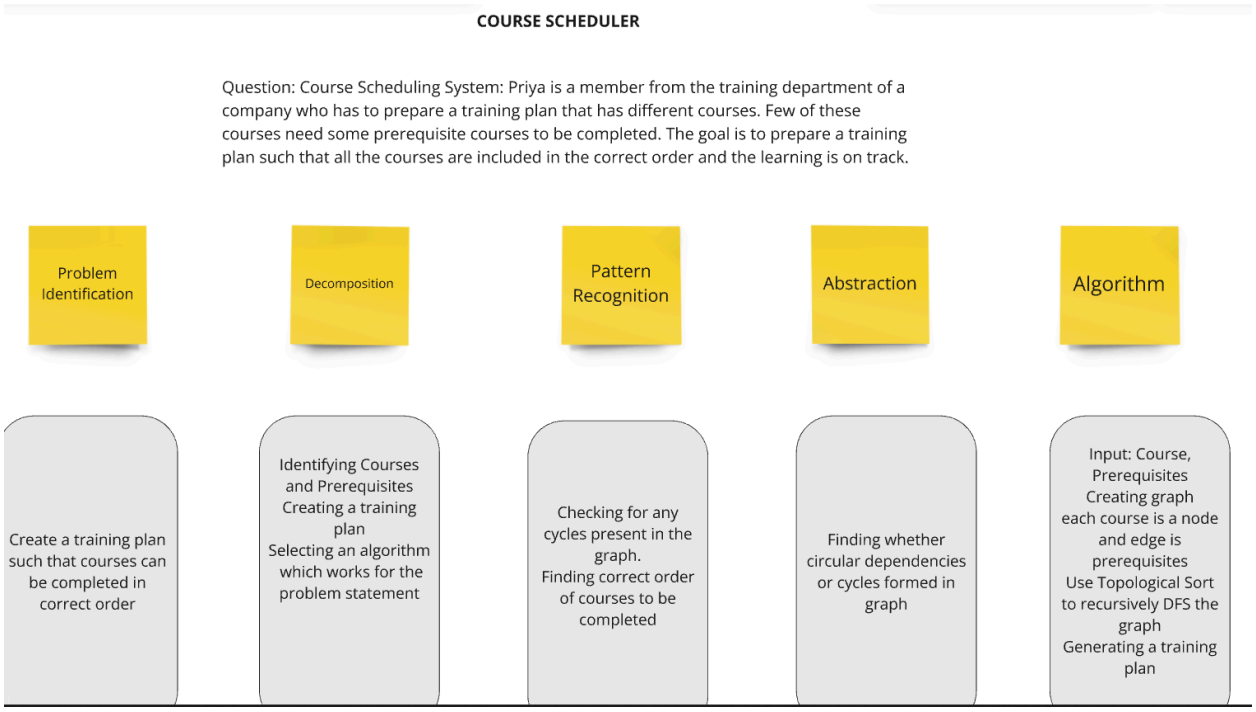
As the prerequisite courses are important before starting some courses we first think of a Depth **First Search** Algorithm to solve these types of problems as we need to completely explore a course before starting any new courses.
Also we can use **Topological Sorting** with DFS
Topological Sorting: It is linear ordering of vertices E=(u,v) u comes before v in the ordering
In our case prerequisites come before courses every time.

Algorithm:

Step 1: Create a class “CourseScheduler” for maintaining the course,prerequisites and for creating a training plan
Step 2: Store all_courses in a Set using self.add_courses=Set()
Step 3: Create methods to add courses, detecting cycles and topological sorting .
Step 4: To generate a Learning Plan it recursively checks if there are no Cyclic dependencies such as A is dependent on C
C is dependent on B
B is dependent on A (Cycle is formed) It is impossible to create a plan for these courses, So we print Error and exit the program.
Step 5: If no cyclic dependencies we print in order of completion of courses such that the training is run smoothly

Screenshots:

Breaking down problem statement with help of Computational Thinking Foundations



OUTPUT : If no circular dependencies are formed we get a Recommended Training Plan based on Course Prerequisites.

OUTPUT: If cyclic dependencies are formed, it is impossible to create a schedule .