LAB 1

check\_count = 0  # Biến toàn cục để đếm số lần kiểm tra ràng buộc

def is\_valid\_assignment(assignment, course, time, constraints):

    """

    Kiểm tra xem việc gán thời gian 'time' cho môn học 'course' có hợp lệ.

    assignment: Dictionary lưu các gán hiện tại {course: time}.

    course: Môn học đang xét.

    time: Khoảng thời gian muốn gán.

    constraints: Danh sách các cặp môn học có ràng buộc.

    """

    global check\_count

    check\_count += 1  # Tăng biến đếm

    for c1, c2 in constraints:

        if c1 == course and c2 in assignment and assignment[c2] == time:

            return False

        if c2 == course and c1 in assignment and assignment[c1] == time:

            return False

    return True

def solve\_csp\_util(courses, times, constraints, assignment, solutions):

    """

    Hàm đệ quy để giải bài toán CSP.

    courses: Danh sách các môn học.

    times: Danh sách các khoảng thời gian.

    constraints: Danh sách các cặp môn học có ràng buộc.

    assignment: Dictionary lưu các gán hiện tại.

    solutions: Danh sách các lời giải.

    """

    # Nếu đã gán hết các môn học, lưu lời giải đầu tiên và dừng

    if len(assignment) == len(courses):

        solutions.append(assignment.copy())

        return True  # Tìm thấy lời giải, dừng

    # Chọn môn học chưa được gán

    for course in courses:

        if course not in assignment:

            # Thử gán từng khoảng thời gian

            for time in times:

                if is\_valid\_assignment(assignment, course, time, constraints):

                    assignment[course] = time

                    if solve\_csp\_util(courses, times, constraints, assignment, solutions):

                        return True  # Tìm thấy lời giải ở nhánh này, dừng

                    del assignment[course]  # Quay lui

            return False # Không tìm thấy lời giải cho môn học này ở bất kỳ khoảng thời gian nào

    return False # Đã gán hết các môn học nhưng không tìm thấy lời giải (trường hợp này không xảy ra nếu logic đúng)

def solve\_csp(courses, times, constraints):

    """

    Hàm chính để giải bài toán CSP.

    courses: Danh sách các môn học.

    times: Danh sách các khoảng thời gian.

    constraints: Danh sách các cặp môn học có ràng buộc.

    """

    global check\_count

    check\_count = 0  # Reset biến đếm

    assignment = {}

    solutions = []

    solve\_csp\_util(courses, times, constraints, assignment, solutions)

    return solutions

def print\_solution(solution):

    """

    In một lời giải.

    """

    for course, time in solution.items():

        print(f"Môn {course}: {time}")

def main():

    # Dữ liệu bài toán ban đầu

    # courses\_orig = ['A', 'B', 'C', 'D']

    # times\_orig = ['T1', 'T2', 'T3']

    # constraints\_orig = [('A', 'B'), ('A', 'C'), ('B', 'D'), ('C', 'D')]

    # Dữ liệu bài toán mở rộng

    courses\_extended = ['A', 'B', 'C', 'D', 'E']

    times\_extended = ['T1', 'T2', 'T3', 'T4']

    constraints\_extended = [('A', 'B'), ('A', 'C'), ('B', 'D'), ('C', 'D'), ('A', 'D'), ('B', 'E'), ('C', 'E'), ('D', 'E')]

    print("Bài toán gán lịch thi (mở rộng):")

    print("Môn học:", courses\_extended)

    print("Khoảng thời gian:", times\_extended)

    print("Ràng buộc:", constraints\_extended)

    solutions = solve\_csp(courses\_extended, times\_extended, constraints\_extended)

    print(f"\nTổng số lần kiểm tra ràng buộc: {check\_count}")

    if solutions:

        print(f"\nTìm thấy {len(solutions)} lời giải.")

        print("\nLời giải đầu tiên tìm được:")

        print\_solution(solutions[0])

    else:

        print("\nKhông tìm thấy lời giải.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Bài toán gán lịch thi (mở rộng):

Môn học: ['A', 'B', 'C', 'D', 'E']

Khoảng thời gian: ['T1', 'T2', 'T3', 'T4']

Ràng buộc: [('A', 'B'), ('A', 'C'), ('B', 'D'), ('C', 'D'), ('A', 'D'), ('B', 'E'), ('C', 'E'), ('D', 'E')]

Tổng số lần kiểm tra ràng buộc: 9

Tìm thấy 1 lời giải.

Lời giải đầu tiên tìm được:

Môn A: T1

Môn B: T2

Môn C: T2

Môn D: T3

Môn E: T1