

BÀI TẬP Tuần 06: Graph cơ bản

1 Bài tập

Bài 1. Hỏi có tồn tại hay không graph đơn vô hướng có 6 đỉnh, 10 cạnh và bậc của 5 đỉnh trong số đó là 1, 3, 3, 3, 5?

Bài 2. Hãy xây dựng một graph đồng bậc, còn gọi là graph đều (regular graph) có 2025 đỉnh mà bậc của tất cả các đỉnh là 10? Câu hỏi tương tự graph có 2026 và đồng bậc 11.

Bài 3. Cho bảng ô vuông 4×4 được chia thành 16 ô vuông con. Hai ô vuông được gọi là “liên kết” nếu có thể đặt quân mã vào đó mà chúng tấn công được nhau (nhắc lại về cách đi của quân mã: nhảy giữa hai ô nằm ở hai góc của hình chữ nhật 3×2 hoặc 2×3).

- a) Xây dựng graph G_4 trong đó tập đỉnh ứng với các ô vuông của bảng, tập cạnh ứng với các cặp ô được liên kết nhau. Dánh số các đỉnh của graph G_4 và cho biết bậc của mỗi đỉnh.
- b) Dựa vào đặc điểm của graph G_4 , tính tổng số cách đặt 2 quân mã giống nhau lên bàn cờ 4×4 sao cho chúng tấn công được nhau.
- c) Tính tổng số cách đặt 3 quân mã giống nhau lên bàn cờ 4×4 sao cho có một quân tấn công được hai quân kia. Gợi ý: không thể có 3 quân mã đối mặt tấn công được nhau.

2 Lập trình

Bài 1. Viết một code block để đọc file g1.v2.jl (tải file này từ moodle) và viết các hàm sau đây để xây dựng đồ thị:

- a) Hàm khởi tạo: đầu vào là đường dẫn đến file cần đọc, đầu ra gồm 3 mảng: Vertices, Edges, Weight. Mảng Vertices sẽ gồm các điểm của đồ thị, mảng Edges bao gồm các cạnh của đồ thị (có hướng). Cuối cùng, mảng Weight sẽ bao gồm trọng số của cạnh (Ví dụ trong database được cho là sẽ là nội dung chuyến bay). Optional: Sinh viên có thể lập trình theo hướng đối tượng (OOP).
- b) Hàm kiểm tra 1: Kiểm tra xem trong mảng Vertices có các đỉnh nào bị trùng không, nếu có thì loại các phần tử bị trùng ra khỏi mảng.
- c) Hàm kiểm tra 2: Giả định không tồn tại có 2 chuyến bay có nội dung bay giống nhau. Kiểm tra xem có tồn tại các chuyến bay bị trùng lặp hay không.
- d) Hàm lấy thông tin: Đầu vào sẽ là sân bay xuất đi và sân bay đến. In ra tất cả thông tin chuyến bay liên quan.

Bài 2. Viết một hàm tìm ra chuyến bay dài nhất và chuyến bay ngắn nhất.

Bài 3. Viết một code block in ra tất cả các chuyến bay của một máy bay có mã được cho (ví dụ: Airbus A318).

Bài 4. Viết một code block kiểm tra xem một đồ thị được lưu có thỏa mãn bài toán bắt tay (handshake theorem).

Bài 5. Viết một code block in ra đỉnh trong đồ thị và bậc tương ứng của chúng.

Bài 6. Viết một code block kiểm tra xem một bước đi/đường đi (walk) có hợp lệ hay không?

Bài 7. Viết một code block kiểm tra xem một vết (trail) có hợp lệ hay không? (Không cạnh nào lặp lại)

Bài 8. Viết một code block kiểm tra xem một đường đi (path) có hợp lệ hay không? (không đỉnh nào lặp lại)

Bài 9. Viết một code block kiểm tra xem một đường đi có phải:

- a) Đường đi đóng (closed walk): bắt đầu và kết thúc tại một điểm.
- b) Chu trình (cycle/circuit): là đường đi đóng có ít nhất một cạnh và không có cạnh nào bị lặp trong đường đi.
- c) Chu trình đơn giản (simple cycle/circuit): là một chu trình chỉ đi qua các đỉnh đúng 1 lần trừ điểm đầu và cuối.

Bài 10. Viết một code block kiểm tra xem đồ thị $G' = [V', E']$ có phải đồ thị con của đồ thị $G = [V, E]$ hay không?