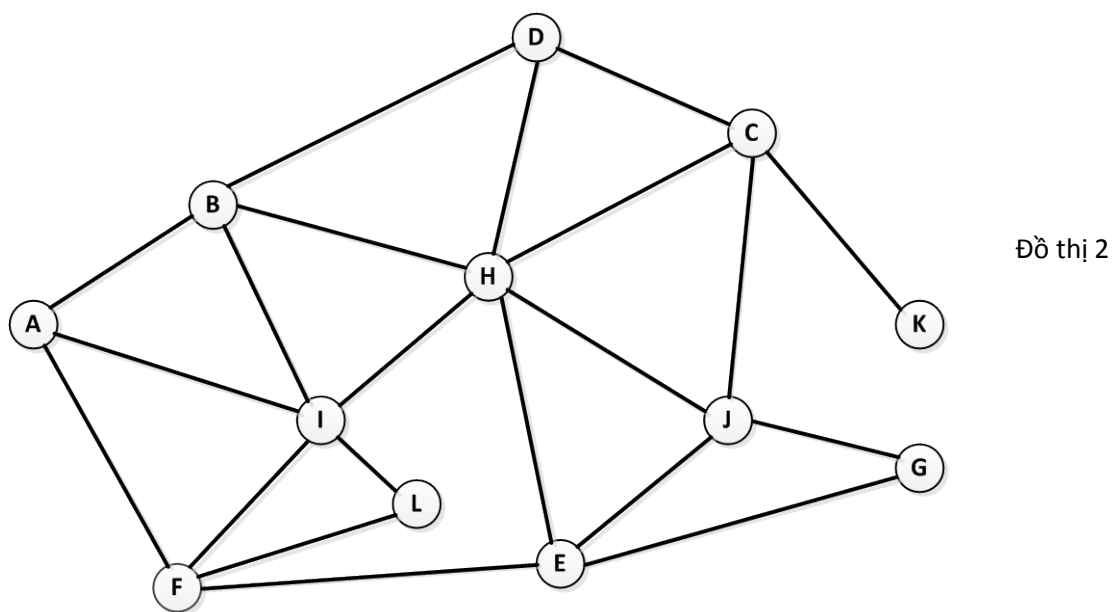
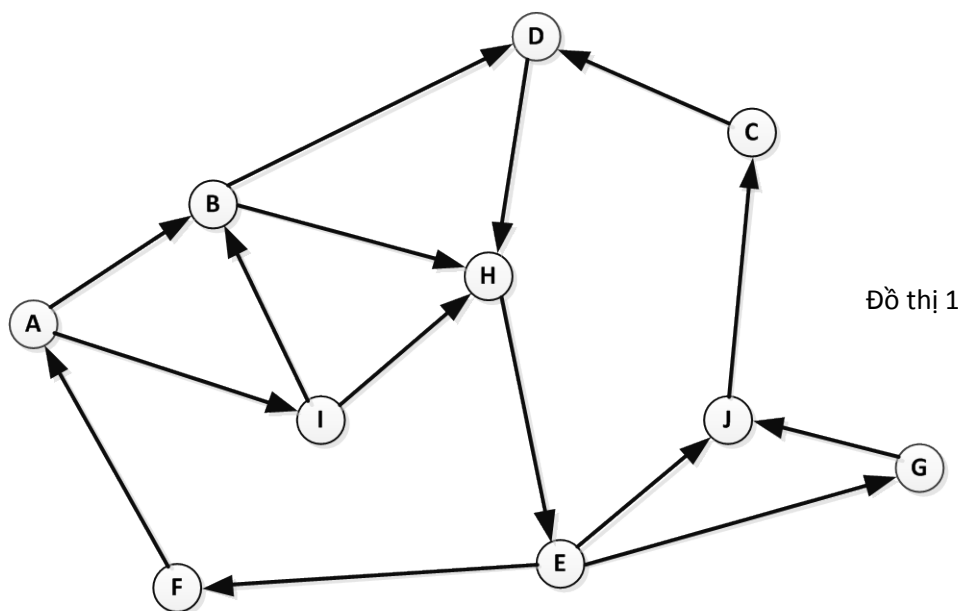


Bài 3. Minh họa cách lưu trữ các đồ thị sau sử dụng ma trận kề và danh sách kề

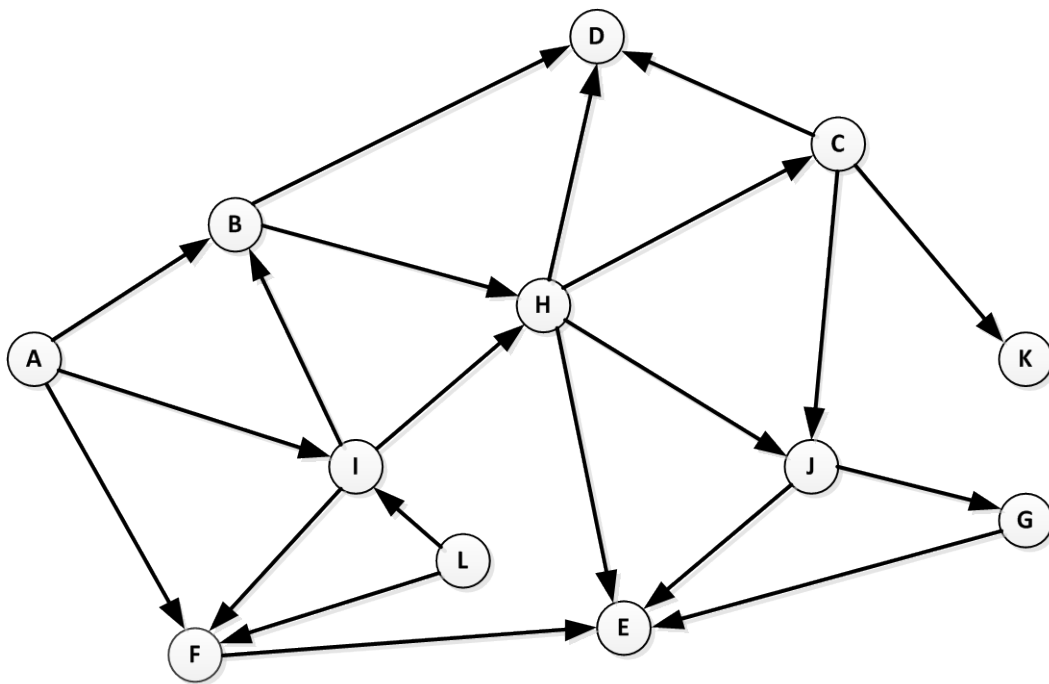
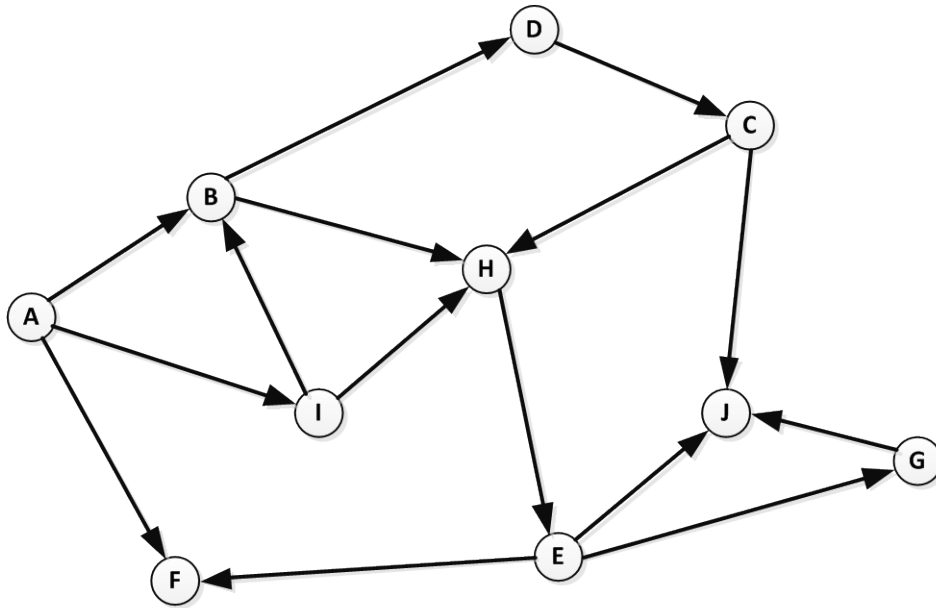


Bài 4. Thực hiện tìm kiếm theo chiều sâu (DFS) và tìm kiếm theo chiều rộng (BFS) trên các đồ thị trong Bài 3 xuất phát từ các đỉnh sau (Đưa ra tất cả các loại cạnh trên cây khung thu được)

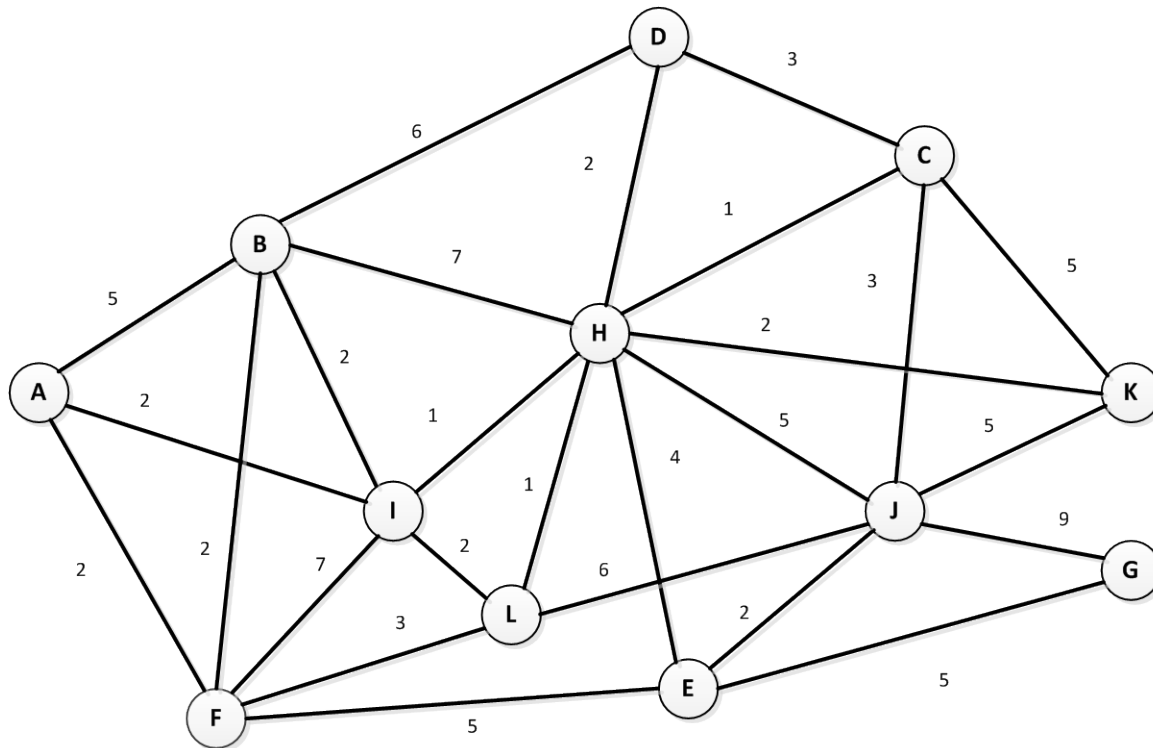
Đồ thị 1: A, B, H

Đồ thị 2: I, B, H

Bài 5. Cho đồ thị có hướng sau, hãy đưa ra một thứ tự topo trên đồ thị đó



Bài 6. Cho đồ thị sau, áp dụng thuật toán PRIM và KRUSKAL để đưa ra cây khung có trọng số nhỏ nhất trên đồ thị (nêu rõ các bước thực hiện)



Bài 7. Đưa ra cây khung có trọng số nhỏ nhất trên đồ thị 1, 2 trong bài 1.

Bài 8. Tìm và đưa ra đường đi ngắn nhất giữa 2 cặp đỉnh (A,K) và (A,G) trên đồ thị trong bài 6

Bài 9. Chứng minh rằng khi duyệt theo chiều sâu (DFS) trên một đồ thị vô hướng thì chỉ có 2 loại cạnh trên cây là Tree edge và Back Edge

Bài 10. Chứng minh bằng quy nạp rằng chỉ có một đường đi duy nhất giữa hai đỉnh bất kỳ trên cây

Bài 11. Chứng minh rằng khi thực hiện BFS trên đồ thị vô hướng $G(V,E)$ thì ta chỉ thu được các cạnh cây (tree edge) và cạnh chéo (cross edge)

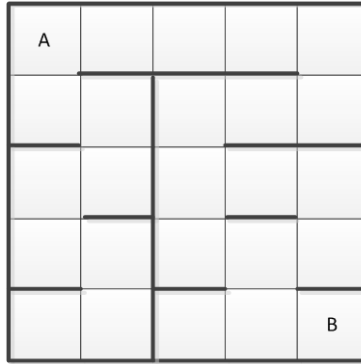
cạnh (u,v) là cross edge nếu u không phải là tổ tiên cũng chẳng phải là con cháu của v

Bài 12. Đơn đồ thị có hướng $G(V,E)$ được lưu trữ bằng ma trận kề. Hãy viết chương trình kiểm tra xem trên đồ thị này có tồn tại chu trình hay không, nếu có thì in ra các chu trình đó.

Bài 13. Cho đơn đồ thị vô hướng $G(V,E)$ được lưu trữ bằng ma trận kề. Hãy viết chương trình kiểm tra xem đồ thị có liên thông hay không. Nếu không liên thông thì hãy chỉ ra đồ thị con liên thông lớn nhất trên G .

Bài 14. Đơn đồ thị $G(V,E)$ được lưu trữ bằng ma trận kề. Viết chương trình tính và in ra bậc của các đỉnh trên đồ thị.

Bài 15. Mô tả thuật toán tìm đường đi trong mê cung



Bài 16. Viết chương trình chuyển đổi các lưu trữ đồ thị từ ma trận kề thành danh sách kề và ngược lại.

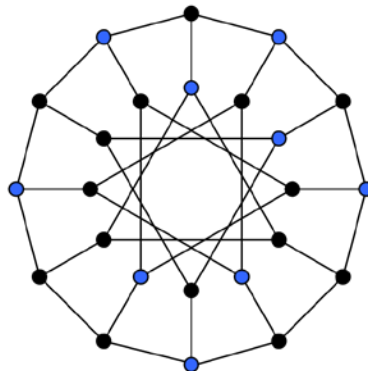
Bài 17. Viết chương trình kiểm tra một danh sách các đỉnh trên đồ thị có phải là một Topological của đồ thị đã cho hay không.

Bài 18. Viết chương trình liệt kê các cây khung nhỏ nhất của đồ thị $G(V,E)$

Bài 19. Tập phủ đỉnh (Vertex cover) của một đồ thị $G(V,E)$ là một tập con các đỉnh $V' \subseteq V$ sao cho mỗi cạnh trên đồ thị đều có ít nhất 1 đỉnh thuộc V' . Hãy viết chương trình tìm tập phủ đỉnh của đồ thị có kích thước nhỏ nhất trong trường hợp $G(V,E)$ là cây.

Bài 20. Tập phủ đỉnh (Vertex cover) của một đồ thị $G(V,E)$ là một tập con các đỉnh $V' \subseteq V$ sao cho mỗi cạnh trên đồ thị đều có ít nhất 1 đỉnh thuộc V' . Nếu ta loại bỏ các nút lá trên cây tìm kiếm thu được khi thực hiện DFS thì các nút còn lại có tạo thành một tập phủ đỉnh của $G(V,E)$? Hãy chứng minh đúng hoặc đưa ra phản ví dụ.

Bài 21. Tập độc lập (independent set) của đồ thị vô hướng $G(V,E)$ là tập các đỉnh U và không cạnh nào trên E có hai đỉnh thuộc U (tức là không có 2 đỉnh nào trong U là kề nhau).



Các đỉnh màu xanh thuộc tập độc lập trên đồ thị

- Viết chương trình tìm tập độc lập lớn nhất (có nhiều đỉnh nhất) trên đồ thị $G(V,E)$ trong trường hợp G là một cây.

Bài 22. Cho đồ thị vô hướng $G(V,E)$, hãy viết chương trình kiểm tra xem trên đồ thị này có tồn tại chu trình kích thước 3 không (chu trình tam giác - triangle).

Bài 23. Cho hai dãy là kết quả của duyệt cây nhị phân theo thứ tự giữa và thứ tự trước, liệu ta có thể xây dựng lại cây ban đầu từ hai dãy này không? Nếu có hãy mô tả thuật toán để xây dựng, ngược lại hãy đưa ra phản ví dụ.

Trong trường hợp chúng ta có thứ tự duyệt trước và duyệt sau thì ta có thể xây dựng lại được cây hay không?

Bài 24. Đưa ra một thuật toán hiệu quả để thực hiện chuyển đổi giữa các mô hình biểu diễn đồ thị sau:

- Chuyển từ ma trận kề sang danh sách kề
- Chuyển từ danh sách kề sang ma trận liên thuộc đỉnh cạnh (incidence matrix). Ma trận liên thuộc đỉnh cạnh là ma trận M của các đỉnh và các cạnh, trong đó $M[i, j] = 1$ nếu đỉnh i là một phần của cạnh j (là nút đầu hoặc nút cuối), và bằng 0 nếu ngược lại.
- Chuyển từ incidence matrix sang danh sách kề.

Đánh giá hiệu quả của thuật toán mà bạn đề xuất.

Bài 25. Xây dựng đồ thị từ các địa điểm chính trong thành phố Hà Nội. Nhập vào hai địa điểm bất kỳ, hãy đưa ra đường đi ngắn nhất giữa hai điểm đó.

Bài 26. Viết hàm duyệt cây nhị phân tìm kiếm và tra về phần tử thứ i theo thứ tự sắp xếp trên cây.

Bài 27. Một đỉnh v trên đồ thị có hướng $G(V, E)$ được gọi là đỉnh mẹ - mother vertex nếu tất cả các đỉnh khác đều có thể tới được từ v (tồn tại đường đi có hướng từ v tới mọi đỉnh trên G)

- Hãy đưa ra một thuật toán để kiểm tra xem v có phải là đỉnh mẹ trên đồ thị có hướng $G(V, E)$. Thời gian thực hiện của thuật toán của bạn cỡ $O(|V| + |E|)$
- Hãy đưa ra một thuật toán để kiểm tra xem trên đồ thị có tồn tại đỉnh mẹ hay không. Thuật toán của bạn cần có thời gian cỡ $O(|V| + |E|)$.

Bài 28. Việc thêm một cạnh có hướng vào đồ thị có hướng $G(V, E)$ có thể làm giảm số lượng thành phần liên thông yếu trên đồ thị. Số lượng thành phần liên thông yếu nào có thể giảm tối đa là bao nhiêu? Số lượng thành phần liên thông mạnh là bao nhiêu?

Bài 29. Giả sử bạn cần sắp xếp cho n bệnh nhân trong một hàng dọc, mặt hướng về đằng trước. Bạn có một danh sách các mối quan hệ của 2 cặp bệnh nhân bất kỳ (i, j) dạng “ i ghét j ”. Nếu i ghét j thì bạn sẽ không thể xếp i đứng sau j được, bởi vì i có khả năng sẽ làm gì đó với j từ đằng sau.

- Đưa ra một thuật toán để xếp hàng (hoặc kết luận là không thể xếp hàng được) với thời gian cỡ $O(|V| + |E|)$
- Giả sử thay vì xếp các bệnh nhân trong một hàng, bạn muốn xếp các bệnh nhân theo nhiều hàng sao cho nếu i ghét j thì i phải ở hàng sau j . Đưa ra một thuật toán hiệu quả để đưa ra số lượng hàng nhỏ nhất này.

Bài 30. Trường hợp bài toán tìm đường đi có trọng số nhỏ nhất giữa hai đỉnh trên đồ thị $G(V, E)$, mà trên đồ thị tồn tại cạnh có trọng số âm.

- Nếu ta vẫn áp dụng thuật toán Dijkstra thì có tìm được đường đi có trọng số nhỏ nhất không? Hãy đưa ra một phản ví dụ nếu câu trả lời là không.
- Nếu ta cải tiến đồ thị bằng cách chuyển hết trọng số âm về trọng số dương (Ví dụ nếu trên đồ thị các cạnh có trọng số là 2, 3, -2 thì ta sẽ chuyển về bộ trọng số tương ứng là 5, 6, 1), và áp dụng Dijkstra để tìm thì có được không? Giải thích tại sao?

Bài 31. Hãy đưa ra thuật toán để tìm đỉnh bắt đầu thực hiện DFS sao cho cây khung thu được có chiều cao là nhỏ nhất.

Bài 32. Hãy đưa ra ý tưởng để thực hiện DFS sao cho cây khung thu được có chiều sâu lớn nhất có thể.

Bài 33. So sánh hiệu quả của PRIM và KRUSKAL khi thực hiện tìm cây khung có trọng số nhỏ nhất? Trong trường hợp đồ thị tổng quát thì nên sử dụng thuật toán nào?

Bài 34. Cài đặt thuật toán tìm cây khung có trọng số lớn nhất

Bài 35. Có khi nào thuật toán PRIM và Kruskal lại cho ta hai cây khung khác nhau hay không? Cho ví dụ?

Bài 36. Đường đi giữa hai đỉnh trên cây khung có trọng số nhỏ nhất – MST tìm được trên đồ thị đầy đủ có phải là đường đi ngắn nhất giữa hai đỉnh đó trên đồ thị? Chứng minh đúng hoặc đưa ra phản ví dụ.

Bài 37. Giả sử rằng tất cả các cạnh trên đồ thị có trọng số khác nhau. Đường đi giữa hai đỉnh bất kỳ trên MST có phải là đường đi ngắn nhất giữa hai đỉnh đó trên đồ thị hay không? Chứng minh đúng hoặc đưa ra phản ví dụ.

Bài 38. PRIM và Kruskal có làm việc được với đồ thị có trọng số âm không? Giải thích?

Bài 39. Gọi T là MST trên đồ thị $G(V, E)$, ta tạo ra đồ thị G' từ G bằng cách cộng các trọng số của các cạnh trên G với một giá trị k . Hỏi T còn là MST trên đồ thị mới G' không? Chứng minh hoặc đưa ra phản ví dụ?

Bài 40. Gọi $P = \{u, \dots, v\}$ là đường đi có trọng số nhỏ nhất từ u đến v trên đồ thị $G(V, E)$. Xây dựng đồ thị G' từ G bằng cách cộng thêm trọng số các cạnh trên G một giá trị k . Hỏi P còn là đường đi ngắn nhất giữa u, v trên đồ thị G' không? Chứng minh đúng hoặc đưa ra phản ví dụ.

Bài 41. Xem xét bài toán tìm tập cạnh có trọng số nhỏ nhất nối các đỉnh trong tập T trên đồ thị $G(V, E)$ ($T \subseteq V$).

- Bài toán này có thể chuyển về bài toán MST được không?
- Hãy đưa ra một thuật toán hiệu quả để tìm tập cạnh có tổng trọng số nhỏ nhất mà kết nối tất cả các đỉnh thuộc T

Bài 42. Hãy sửa đổi lại thuật toán PRIM để nó thực hiện với thời gian $O(n \log k)$ trên một đồ thị có k loại trọng số khác nhau

Bài 43. Bài toán single-destination shortest path trên đồ thị có hướng tìm kiếm tất cả đường đi ngắn nhất từ tất cả các đỉnh tới một đỉnh v đặc biệt. Hãy đưa ra một thuật toán hiệu quả để giải bài toán này.

Bài 44. Cho đồ thị vô hướng có trọng số $G(V, E)$, và T là cây khung đường đi ngắn nhất có gốc là đỉnh v

- Hãy đưa ra một đồ thị mà MST trên đồ thị trùng với cây khung đường đi ngắn nhất có gốc là v
- Hãy đưa ra một đồ thị mà MST trên đồ thị khác với cây khung đường đi ngắn nhất có gốc là v