

Cấu Trúc Dữ Liệu & Giải Thuật

Data Structures & Algorithms

GV: Phan Hồng Trung

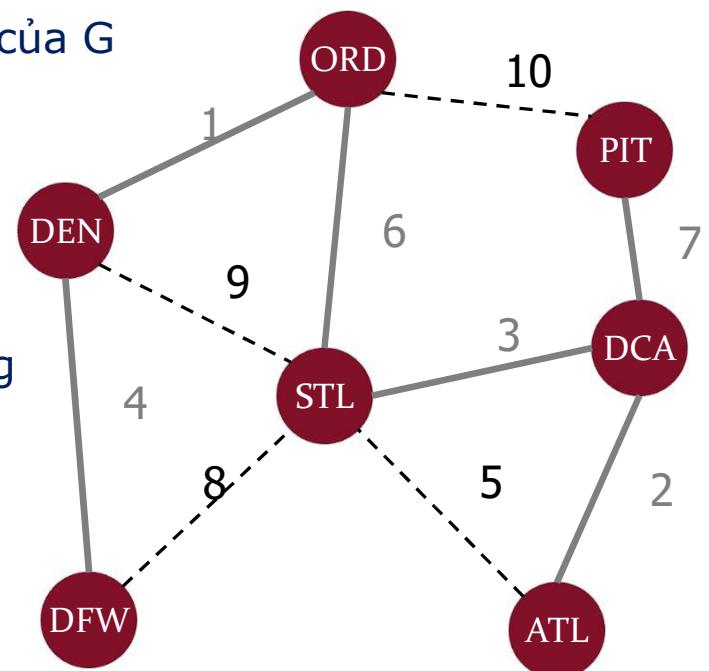
Bài 10 – Đồ Thị (Graph) – Phần 2

- Cây phủ:
 - Thuật toán Prim-Jarnik
 - Thuật toán Kruskal
- Đồ thị Euler



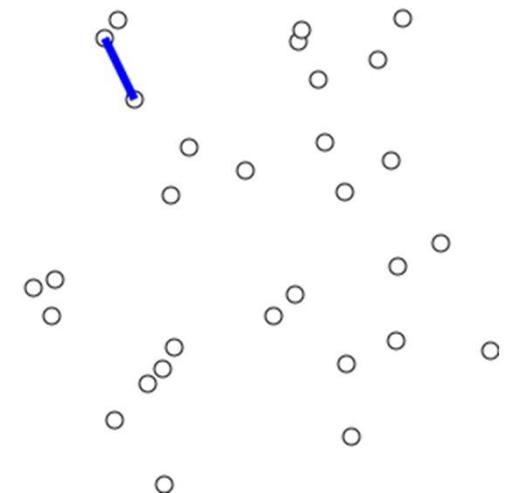
Cây phủ tối thiểu

- Đồ thị con phủ:
 - Đồ thị con của đồ thị G chứa tất cả các đỉnh của G
- Cây phủ:
 - Đồ thị con phủ là một cây
- Cây phủ tối thiểu (MST):
 - Cây phủ của đồ thị có trọng số với tổng trọng số cạnh nhỏ nhất
- Ứng dụng:
 - Mạng lưới truyền thông
 - Mạng lưới giao thông

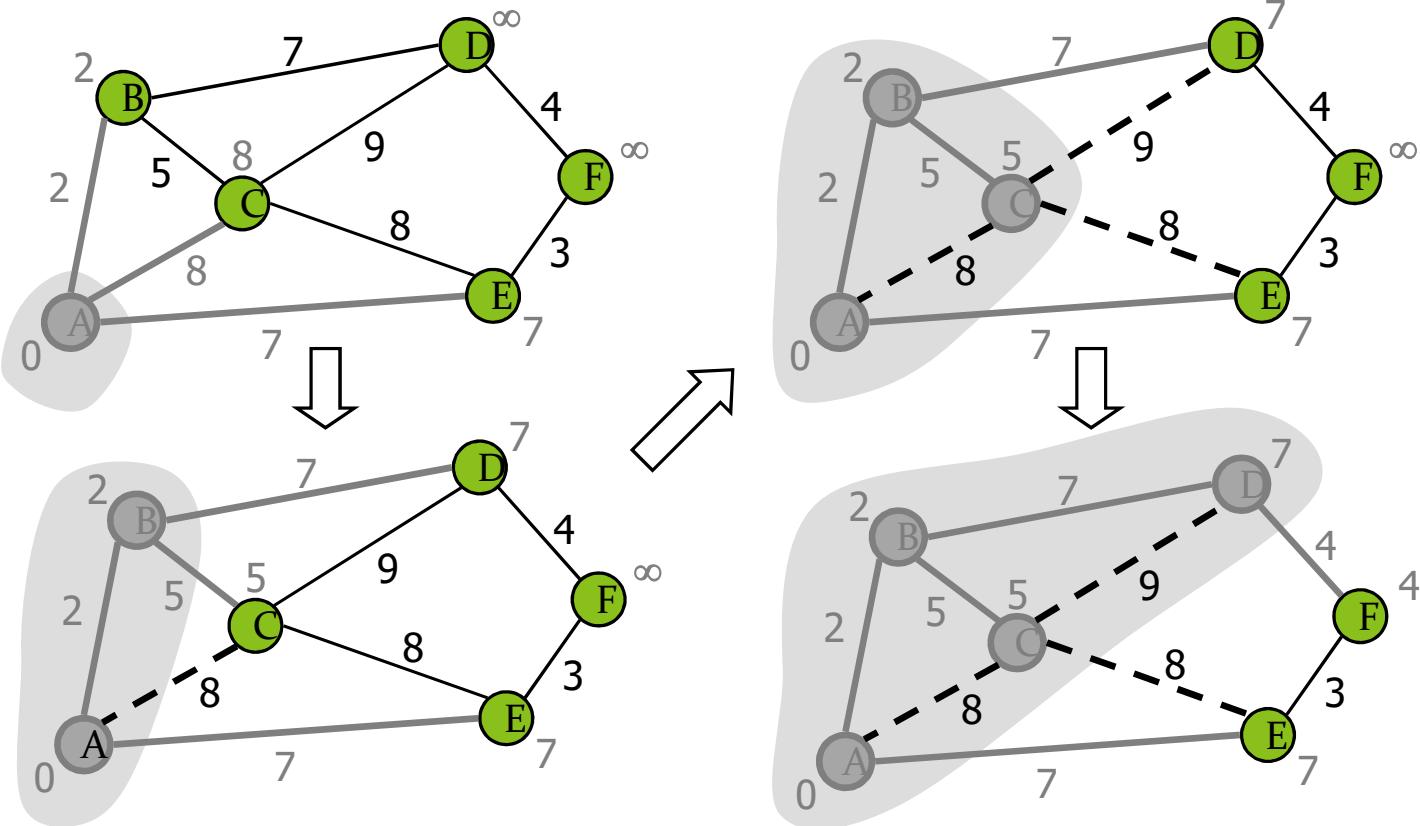


Thuật toán Prim-Jarnik

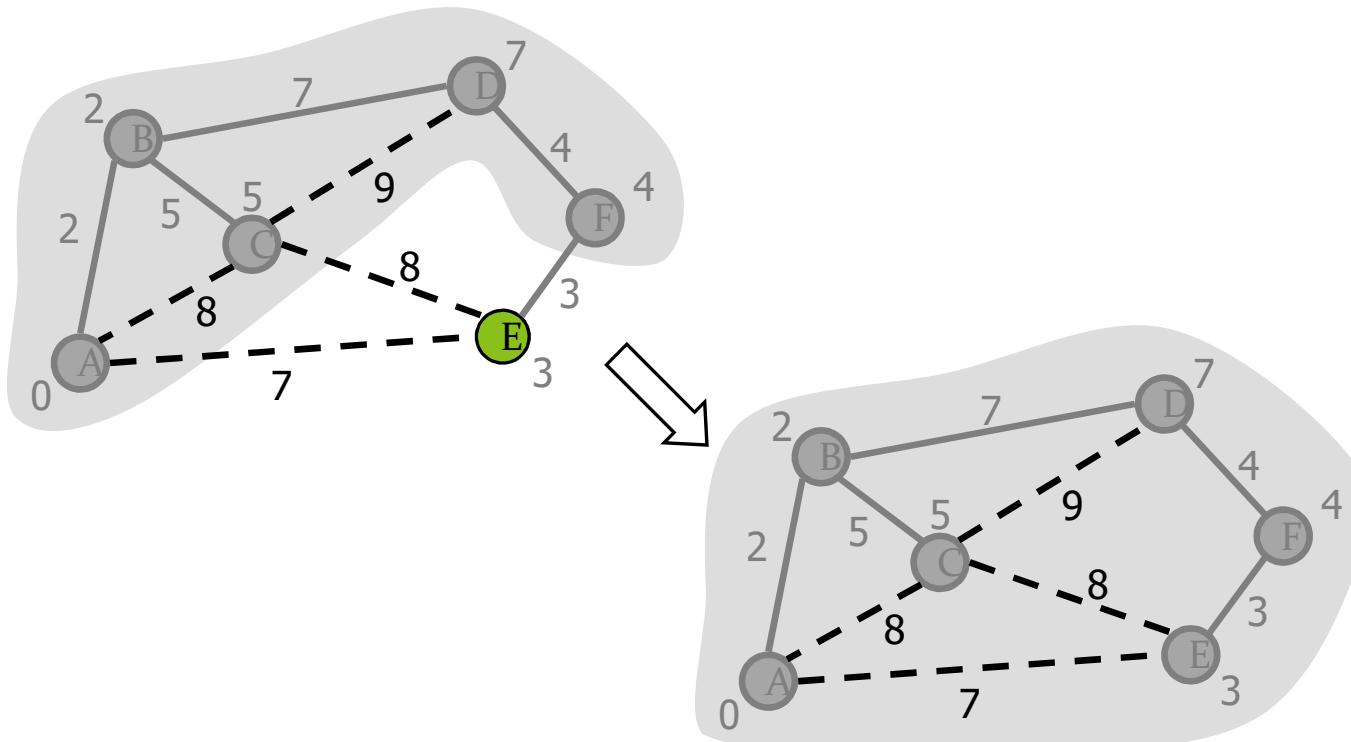
1. Khởi tạo cây với một đỉnh duy nhất, được chọn tùy ý từ đồ thị.
2. Thêm một cạnh có **trọng số nhỏ nhất** vào cây để nối cây với một đỉnh chưa có trong cây.
3. Lặp lại bước 2 (cho đến khi tất cả các đỉnh đều có trong cây).



Thuật toán Prim-Jarnik – Ví dụ

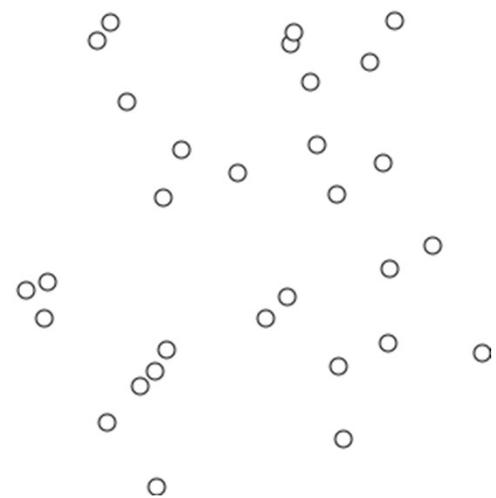


Thuật toán Prim-Jarnik – Ví dụ

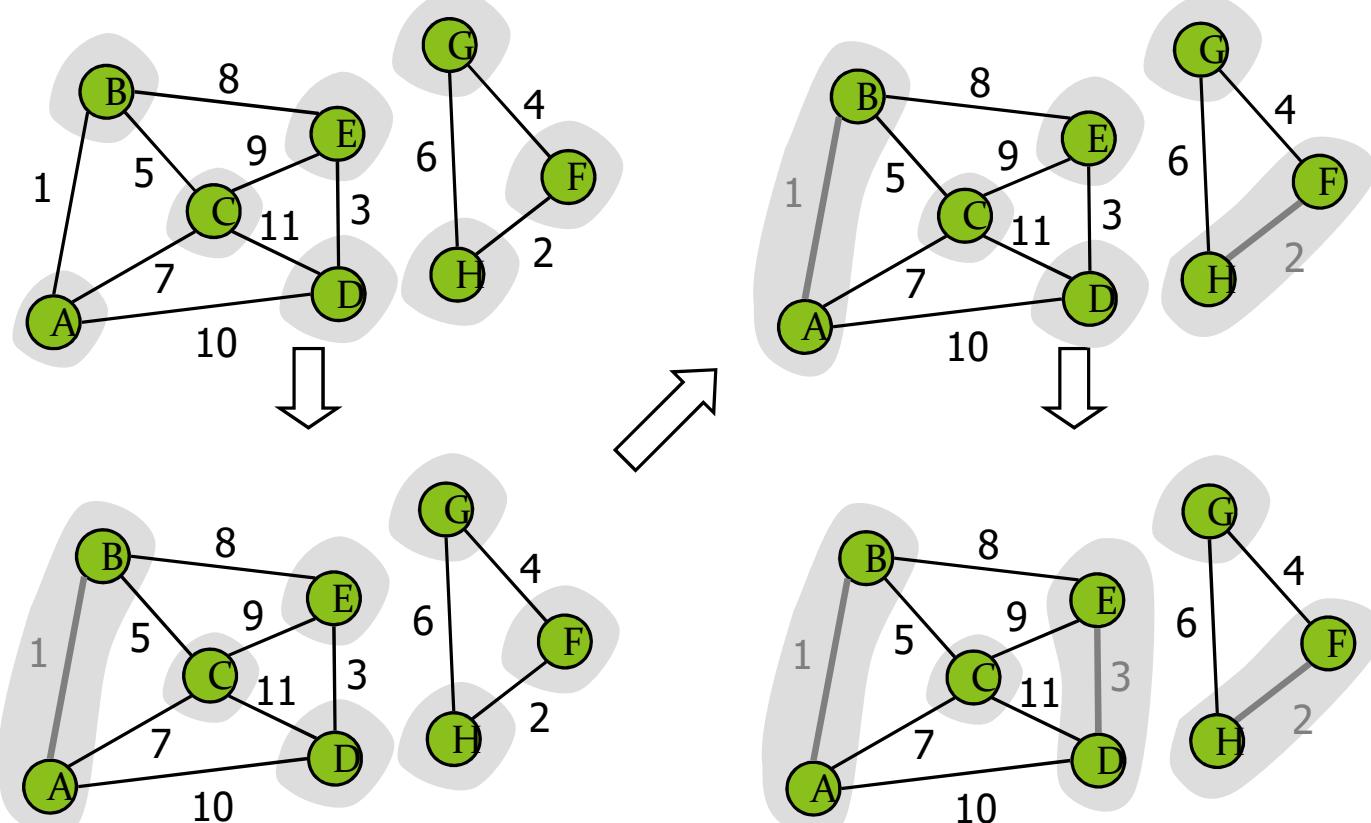


Thuật toán Kruskal

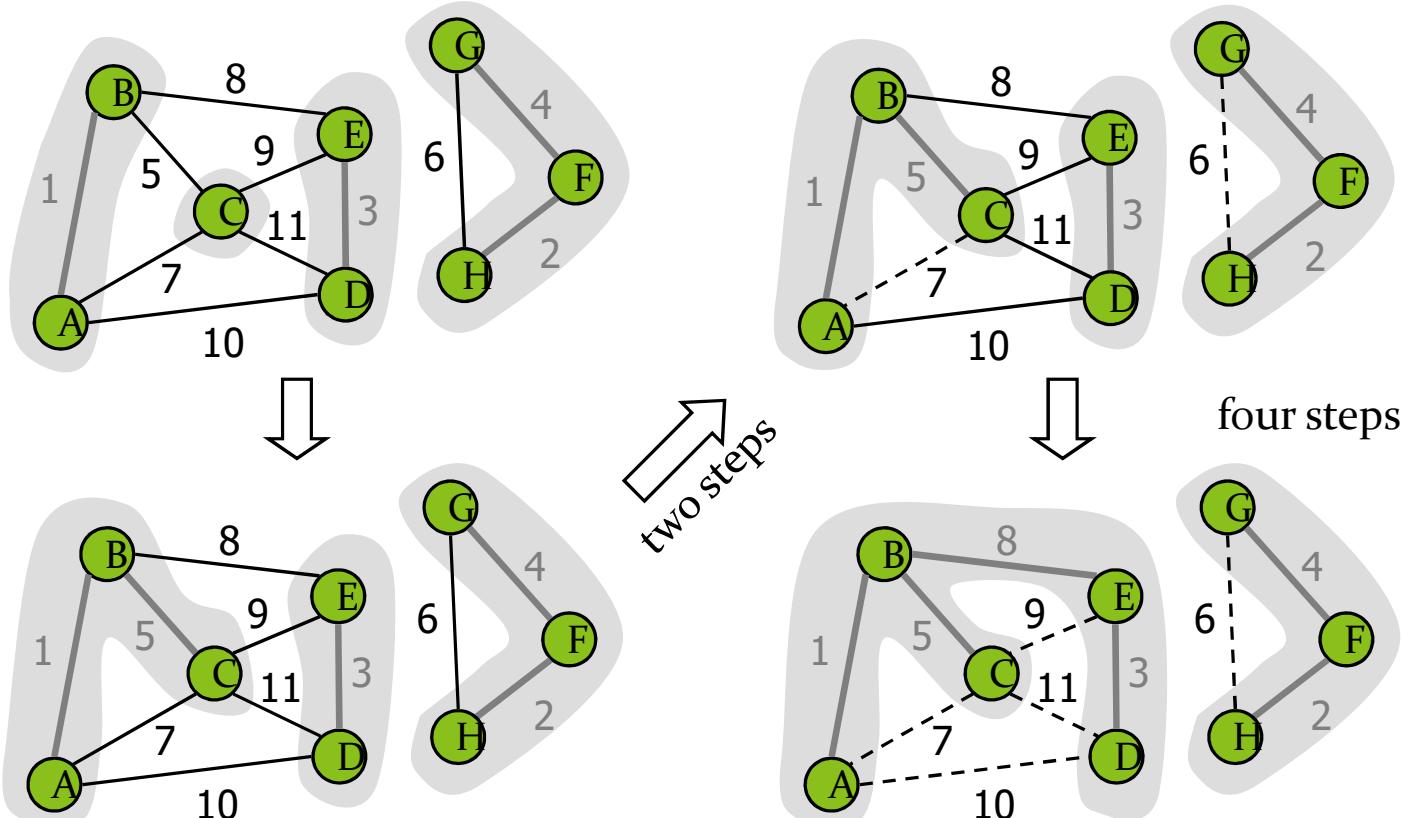
- Tạo một rừng F (một tập hợp các cây), trong đó mỗi đỉnh trong đồ thị là một cây riêng biệt
- Tạo một tập hợp S chứa tất cả các cạnh trong đồ thị
- Trong khi S khác rỗng và F chưa mở rộng:
 - Xóa một cạnh có trọng số nhỏ nhất khỏi S
 - Nếu cạnh bị xóa kết nối hai cây khác nhau thì thêm nó vào rừng F, kết hợp hai cây thành một cây duy nhất
- Khi thuật toán kết thúc, rừng tạo thành một rừng cây phủ tối thiểu của đồ thị. Nếu đồ thị được kết nối, rừng có một cây duy nhất, chính là cây phủ tối thiểu của đồ thị.



Kruskal Algorithm – Example

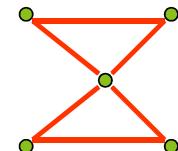


Kruskal Algorithm – Example

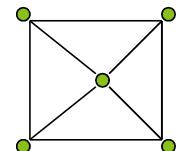


Chu trình và đường đi Euler

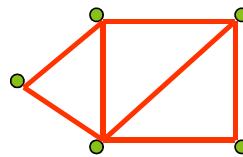
- Đường đi Euler: đường đi qua tất cả các cạnh của đồ thị đúng một lần.
- Chu trình Euler: chu trình đi qua tất cả các cạnh của đồ thị đúng một lần.



Has Euler cycle

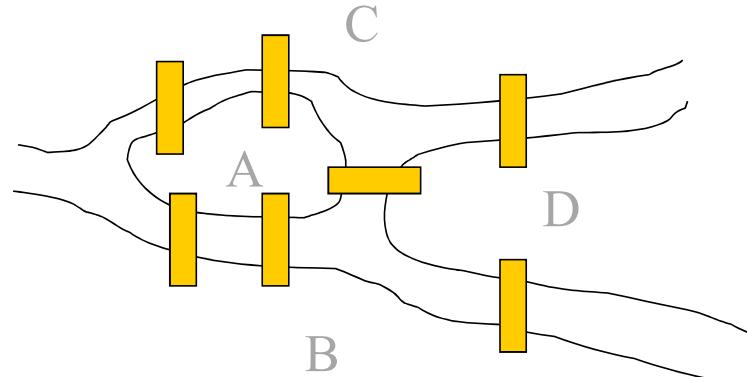


No Euler cycle
No Euler path



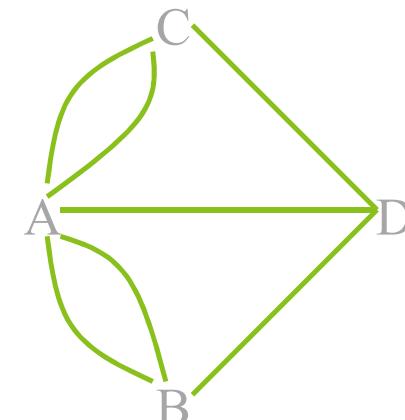
Has Euler path
No Euler cycle

Những cây cầu ở Königsberg



- Đảo Kniphofia trên sông Pregel và 7 cây cầu được xây dựng vào thế kỷ 18 tại thị trấn Königsberg (Kalingingrad).

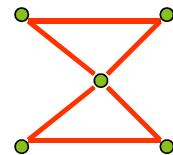
- Có thể bắt đầu ở một địa điểm bất kỳ, đi qua tất cả cây cầu mà không phải đi qua bất kỳ cây cầu nào hai lần và quay trở lại địa điểm bắt đầu không?
- Diễn đạt lại theo chu trình Euler:



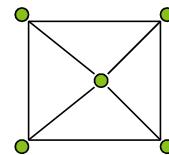
Điều kiện có chu trình Euler

■ Định lý 1:

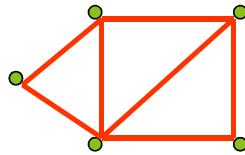
- Một đa đồ thị liên thông có chu trình Euler khi và chỉ khi mỗi đỉnh của nó có bậc chẵn.



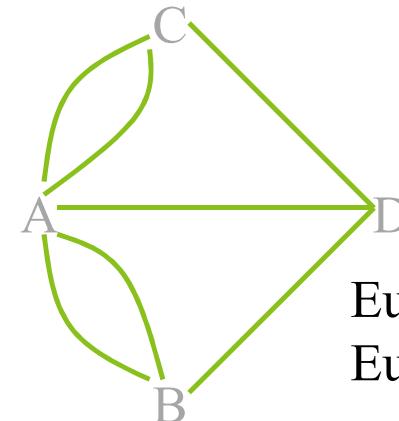
Has Euler cycle



No Euler cycle
no Euler path



Has Euler path,
but no Euler cycle

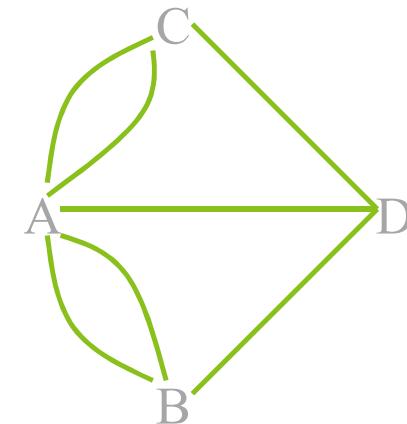
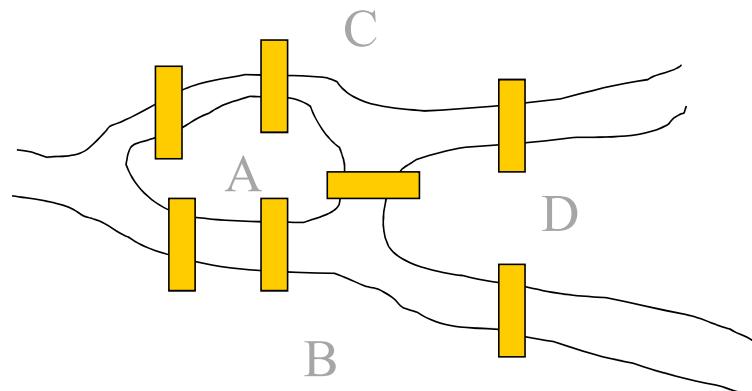


Euler cycle?
Euler path?

Điều kiện có đường đi Euler

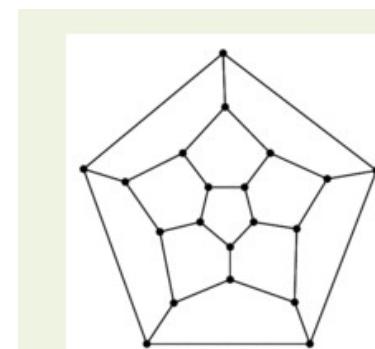
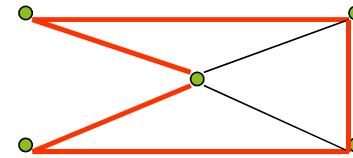
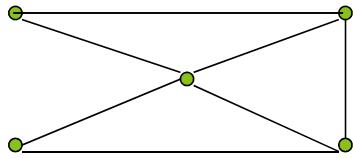
■ Định lý 2:

- Một đa đồ thị liên thông có đường đi Euler nhưng không có chu trình Euler khi và chỉ khi nó có đúng hai đỉnh bậc lẻ.

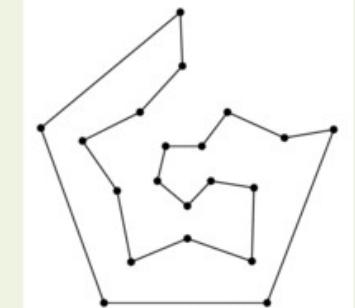


Đường đi và chu trình Hamilton

- Chu trình Hamilton: chu trình đi qua mọi đỉnh của đồ thị đúng một lần.
- Đường đi Hamilton: đi qua mọi đỉnh của đồ thị đúng một lần.

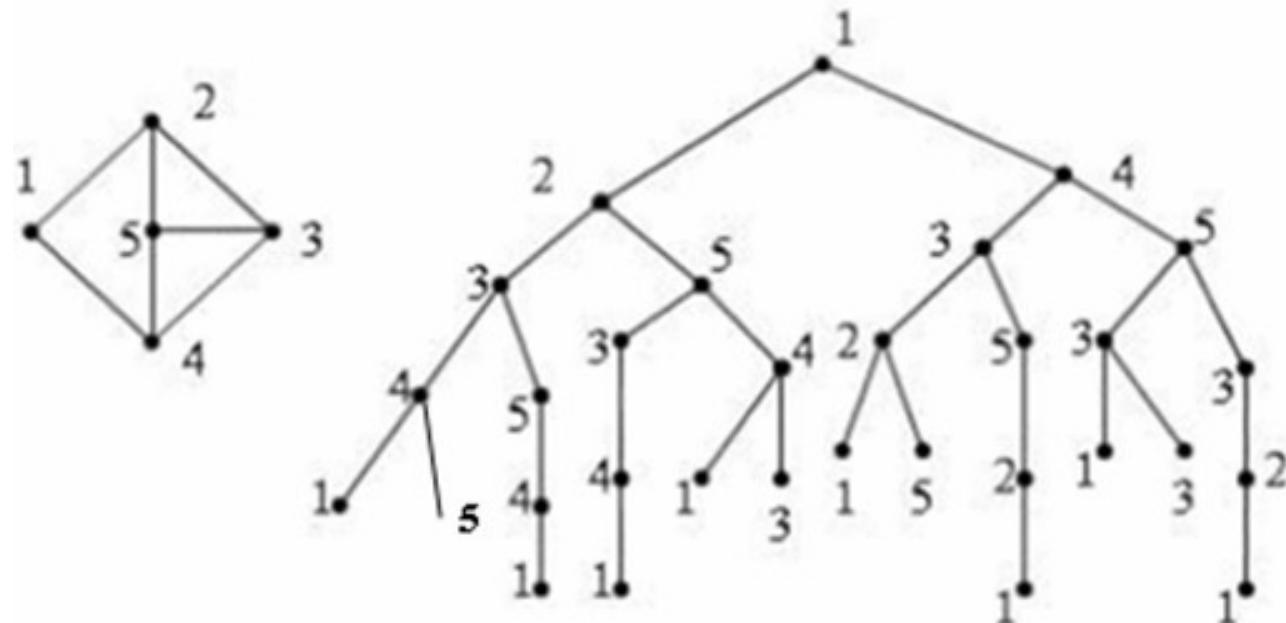


INPUT



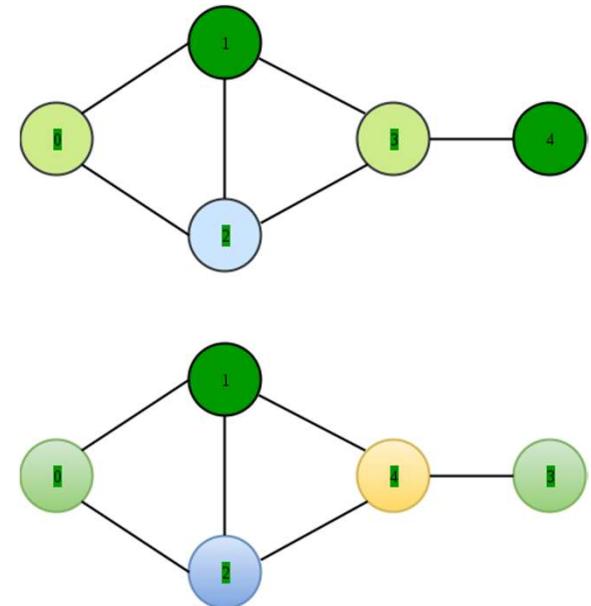
OUTPUT

Dùng Backtracking liệt kê chu trình Hamilton



Tô màu đồ thị

- Tô màu đồ thị là một cách tô màu các đỉnh của đồ thị sao cho không có hai đỉnh kề nhau có cùng màu.
- Ứng dụng:
 - Tô màu bản đồ, lập lịch, khớp mẫu, lập lịch thể thao, thiết kế sơ đồ chỗ ngồi, lập thời gian biểu thi, lập lịch taxi và giải câu đố Sudoku.



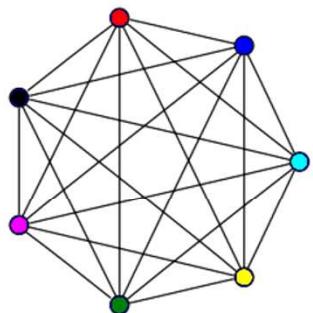
Tô màu đồ thị

- Số sắc tố của đồ thị là số màu tối thiểu mà có thể sử dụng để tô màu các đỉnh của đồ thị sao cho không có hai đỉnh kề nhau có cùng màu.
- Số sắc tố của đồ thị G được ký hiệu là $\chi(G)$ (Chi /'kai/ của G).
- Một đồ thị mà $k = \chi(G)$ được gọi là k -colorable.

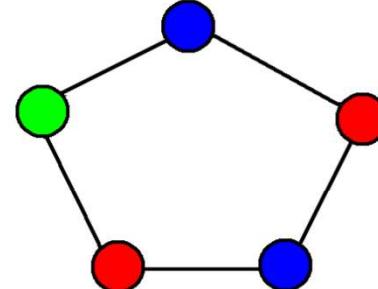


Tô màu đồ thị

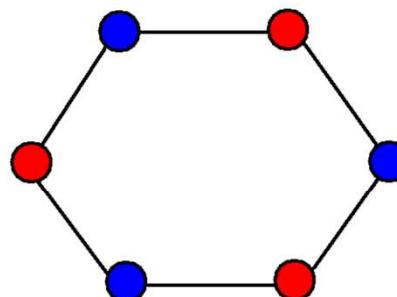
- Đồ thị đầy đủ (complete graphs) $\chi(K_n)=n$
- Đồ thị chu trình (cycle graphs) $\chi(C_{2n})=2$, $\chi(C_{2n+1})=3$
- Đồ thị lưỡng cực (bipartite graphs) $\chi(K_{n,n})\leq 2$.



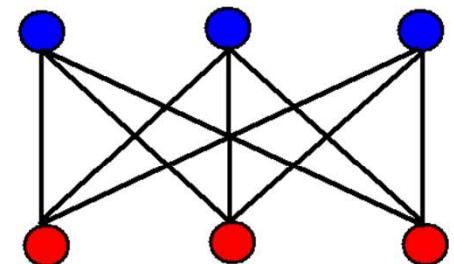
$$\chi(K_7) = 7$$



$$\chi(C_5) = 3$$



$$\chi(C_6) = 2$$



$$\chi(K_{3,3}) = 2$$



