Cây (Tree)

Cây bao trùm (spanning tree)

- Cây bao trùm (cây khung) của một đơn đồ thị G là một đồ thị con của G, chứa tất cả các đỉnh của G.
  - Một đơn đồ thị có cây bao trùm thì liên thông.
  - Một đơn đồ liên thông thì có cây bao trùm.

- **Định lý**: Một đơn đồ thì liên thông nếu và chỉ nếu nó có cây bao trùm.

Cây bao trùm

- Có thể loại bỏ các cạnh khỏi các chu trình đơn để tìm cây bao trùm trong một đồ thị
  - Không hiệu quả vì nó yêu cầu phải xác định được các chu trình đơn này.

- Thay vào đó, tìm bằng cách thêm các cạnh sử dụng các thuật toán duyệt đồ thị
  - Theo chiều sâu (Depth-First Search)
  - Theo chiều rộng (Breadth-First Search).

Depth-First Search

Procedure DFS(G: đồ thị liên thông với các đỉnh $v_i, i = 1, \ldots, n$)

T := cây có duy nhất một đỉnh v_1

visit(v_1)

Procedure visit(v: đỉnh của G)

for mỗi đỉnh w kề với v và chưa nằm trong T

Thêm đỉnh w và cạnh {v, w} vào T

visit(w)

Thủ tục DFS xây dựng cây bao trùm với độ phức tạp là O(e) hay $O(n^2)$ trong đó e và n lần lượt là số cạnh và số đỉnh của G

Breadth-First Search

Procedure BFS(G: đồ thị liên thông với các đỉnh $v_i, i = 1, \ldots, n$)

T := cây có duy nhất một đỉnh v_1

L := danh sách rỗng (danh sách các đỉnh chưa được thăm)

Thêm v_1 vào L

while L không rỗng

Lấy đỉnh đầu tiên, v, ra khỏi L

for mỗi lân cận w của v

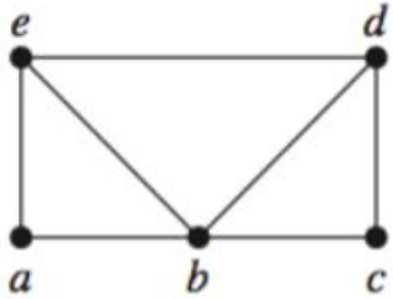
if w không năm trong L và không trong T **then**

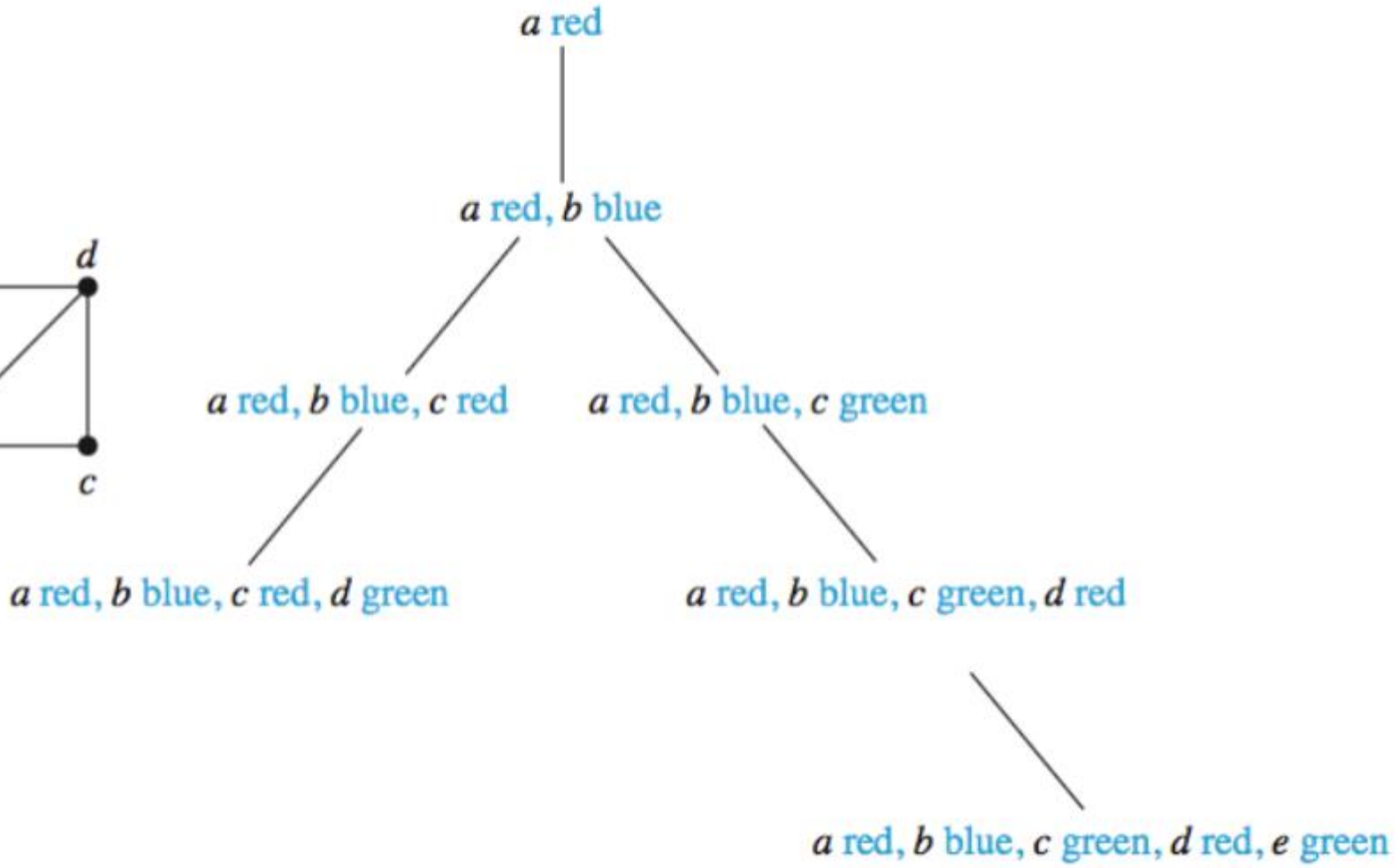
Thêm w vào cuối L

Thêm w và cạnh {v, w} vào T

Độ phức tạp của thuật toán BFS là O(e) hay $O(n^2)$

Cây trong thuật toán quay lui

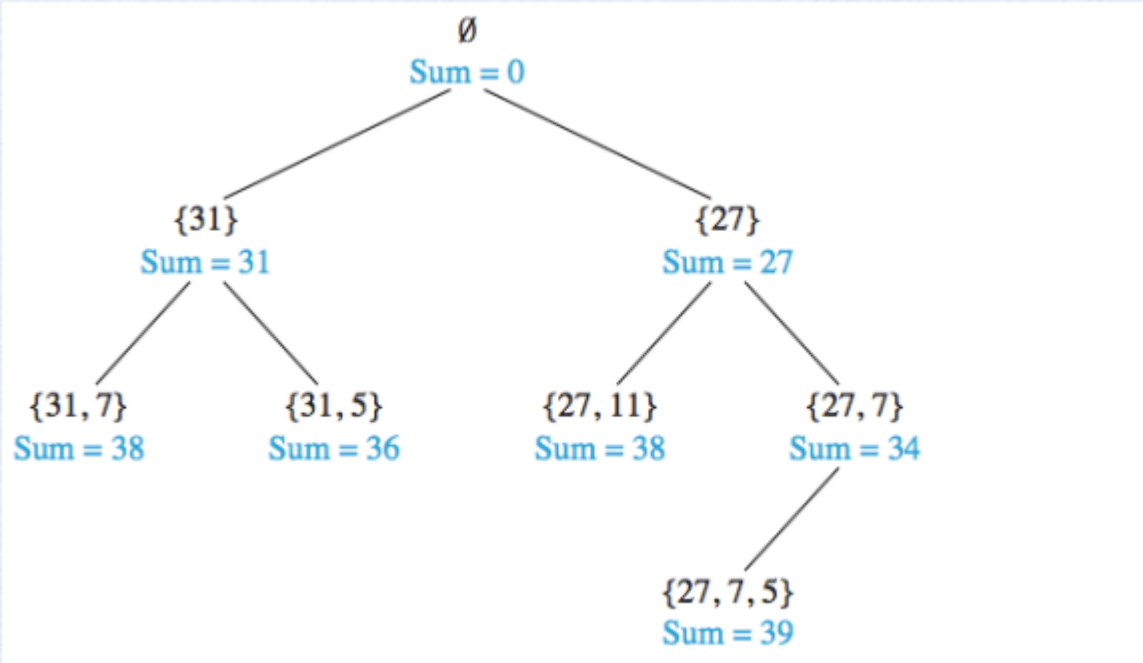




Tô màu đồ thị (bên trái màn hình) sử dụng thuật toán quay lui

Cây trong thuật toán quay lui

- Tổng của tập con: Cho một tập gồm n phần tử x_1, x_2, ... x_n. Tìm một tập con mà tổng các phần tử của nó bằng M.

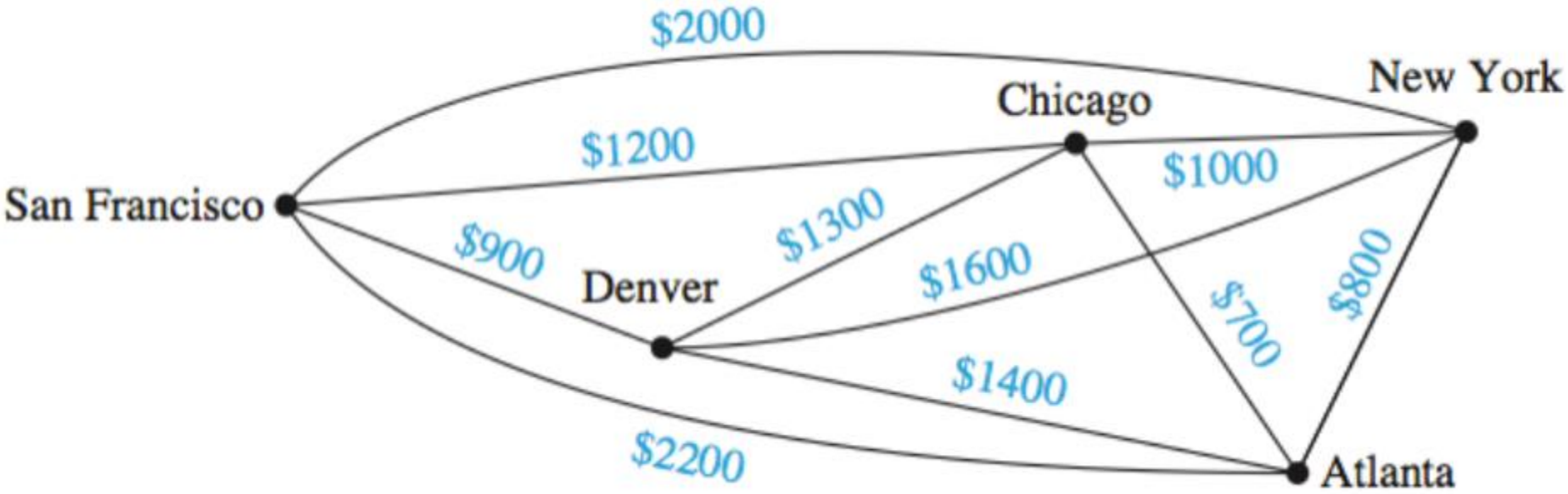


Lược đồ quay lui để tìm các số có tổng bằng 39

DFS trên đồ thị có hướng

- Ta có thể dễ dàng thay đổi 2 thuật toàn DFS và BFS phù hợp với đồ thị có hướng.
- Đầu ra không nhất thiết là một cây bao trùm, mà đúng hơn là một rừng bao trùm
  - Nếu ở một giai đoạn của một trong hai thuật toán chúng ta thấy rằng không có cạnh nào tồn tại từ một đỉnh đã được thêm vào tới một đỉnh chưa được thêm vào, đỉnh tiếp theo được thêm bởi thuật toán trở thành gốc của một cây mới trong rừng bao trùm đó.

Cây bao trùm nhỏ nhất



Một đồ thị có trọng số thể hiện giá thuê bao hàng tháng cho các đường truyền trong một mạng máy tính

Tìm cây bao trùm mà tổng trọng số trên các cạnh của cây này là nhỏ nhất.

Cây bao trùm nhỏ nhất

- Hai thuật toán tham lam đã được đề xuất để tìm cây bao trùm nhỏ nhất:
  - Thuật toán Prim
    - Đề xuất bởi một nhà toán học người Séc vào năm 1930
    - Phổ biến khi được tái nghiên cứu bởi Prim vào năm 1957
  - Thuật toán Kruskal
    - Joseph Kruskal đề xuất vào năm 1956

Thuật toán Prim

Procedure Prim(G: đồ thị vô hướng liên thông có trọng số với n đỉnh)

T := một cạnh có trọng số nhỏ nhất

for i:= 1 to n-2

e:= một cạnh có trọng số nhỏ nhất có gắn với một đỉnh trong T và không tạo một chu trình trong T nếu được thêm vào T

T:= T với e được thêm vào

return T {T là cây khung/bao trùm nhỏ nhất của G}

Thuật toán Kruskal

Procedure Kruskal(G: đồ thị vô hướng liên thông có trọng số với n đỉnh)

T := đồ thị rỗng

for i:= 1 to n-1

e:= một cạnh có trọng số nhỏ nhất không tạo một chu trình trong T nếu được thêm vào T

T:= T với e được thêm vào

return T {T là cây khung/bao trùm nhỏ nhất của G}