ĐƯỜNG ĐI NGẮN NHẤT TRÊN ĐỒ THỊ

- Các khái niệm
- Thuật giải Dijkstra
- Thuật giải Bellman-Ford

CÁC KHÁI NIỆM

- Cho đồ thị G=(V, E) có trọng số, |V|=n, |E|=m
  - Nếu $(u, v) \in E$ thì $w(u, v) = \alpha < \infty$
  - Ngược lại $(u, v) \notin E$ thì coi $w(u, v) = \infty$
  - Trọng số của đường đi $P=v_{0,}\ v_1,\ldots v_k$ là $w(P)=\Sigma_{i=1,\ k} w(v_{i-1}, v_i)$

CÁC KHÁI NIỆM

- Trọng số của đường đi ngắn nhất từ u đến v là

- $d(u, v) = \begin{cases} \min\{w(p)\}, & \text{nếu có đường đi p từ u đến v} \\ \infty & \text{nếu ngược lại} \end{cases}$

- Một đường đi p từ u đến v mà w(p)=d(u, v) gọi là đường đi ngắn nhất từ u đến v (cũng gọi d(u, v) là khoảng cách từ u đến v)

THUẬT GIẢI DIJKSTRA

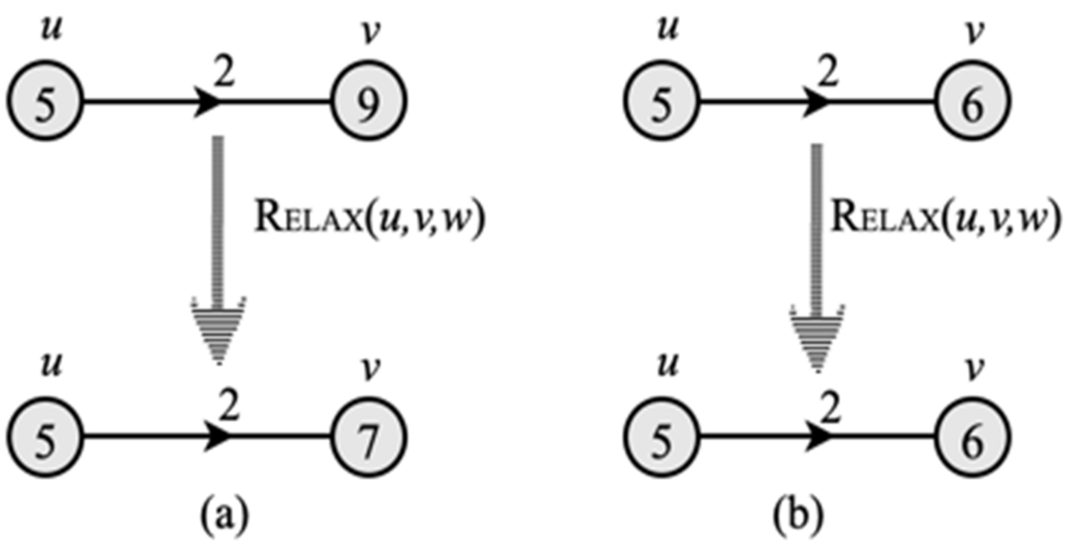
- **Bài toán:** Tìm các đường đi ngắn nhất từ một đỉnh s đến mọi đỉnh khác trong một đồ thị có trọng số không âm

THUẬT GIẢI DIJKSTRA

Ý tưởng

- Ký hiệu d[v] là một cận trên của d(s,v), thuật giải kiểm tra và giảm d[v] cho đến khi d[v]=d(s, v)
  - Nếu d[v]>d[u]+w(u,v) thì làm tốt cận trên d[v] bằng cách gán d[v]= d[u]+w(u,v) (gọi là relaxation)
  - Nếu d[v] đã tốt nhất thì đưa v vào trong tập S = {v $\in$V | d[v] =d(s, v)}, lúc này d[v] là độ dài đường đi ngắn nhất từ s đến v

THUẬT GIẢI DIJKSTRA



Nếu d[v]>d[u]+w(u,v)

thì gán d[v]= d[u]+w(u,v)

THUẬT GIẢI DIJKSTRA

```
RELAX(u, v, w)
if d[v] > d[u] + w(u, v)
    d[v] = d[u] + w(u, v)
    π[v] = u
```

THUẬT GIẢI DIJKSTRA

```
DIJKSTRA(G, w, s)
    INITIALIZE-SINGLE-SOURCE(G, s)
    S = ∅
    Q = V[G]
    while Q ≠ ∅
        u = EXTRACT-MIN(Q)
        S = S∪{u}
        for each v ∈ Adj[u]
            RELAX(u, v, w)
```

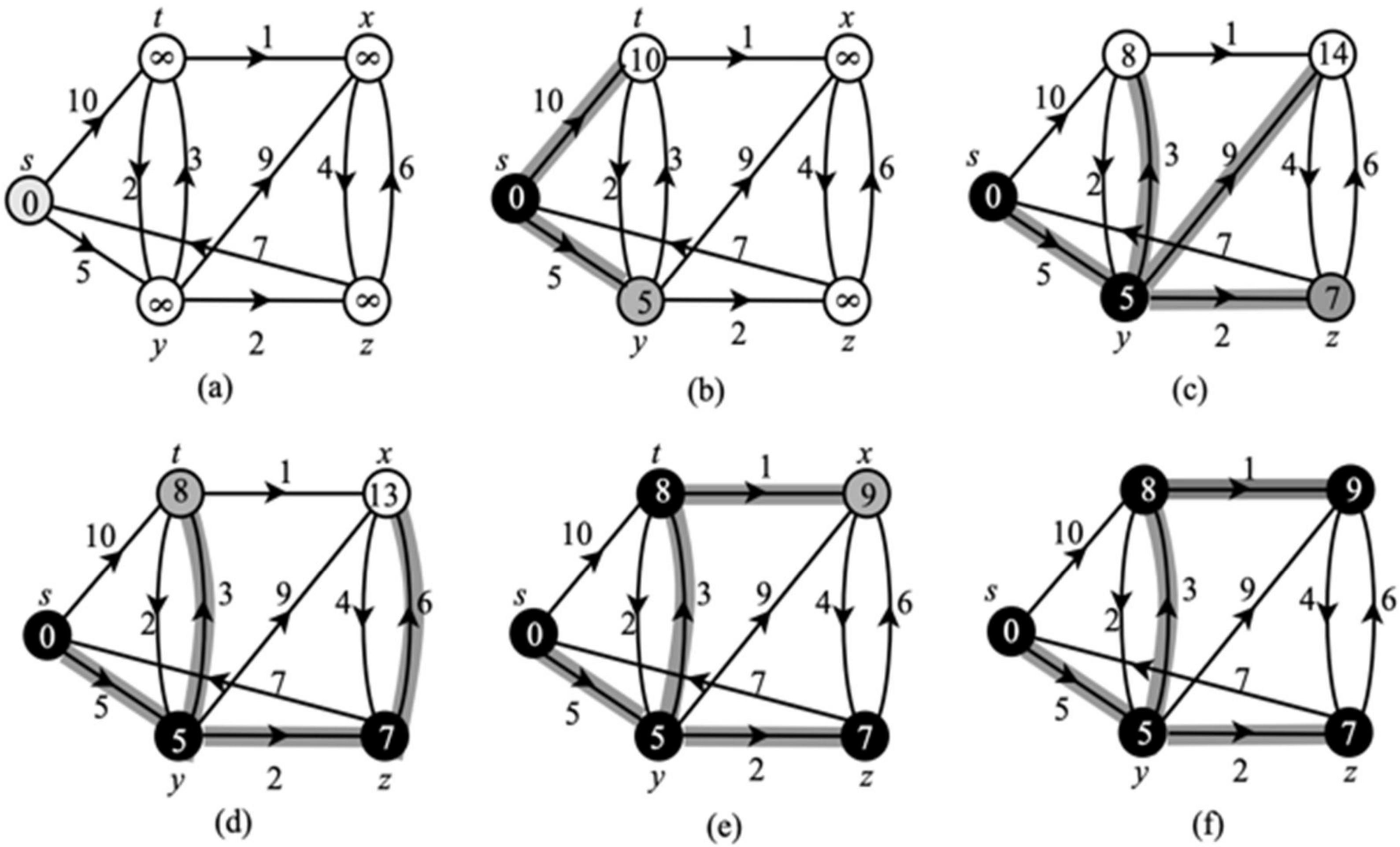
THUẬT GIẢI DIJKSTRA

```
INITIALIZE-SINGLE-SOURCE(G, s)
for each v ∈ V[G]
    d[v] = ∞
    π[v] = NIL
d[s] = 0
```

THUẬT GIẢI DIJKSTRA

- Thời gian chạy từ dòng 1-3 là $O(V)$
- Thời gian thực hiện Extract-Min tại dòng 5 là $O(\log_2 V)$
- Tổng số thao tác Relax tại vòng lặp **for** 7-8 đối với tất cả các đỉnh u được chọn tại dòng 5 là $O(E)$
- Mỗi lần Relax một cạnh (u, v) chi phí là $O(\log_2 V)$
- Số lần lặp của vòng lặp **while** 4-8 là V
- Vậy, chi phí của thuật giải Dijkstra là $O(V) + O((V+E)\log_2 V) = O((V+E)\log_2 V)$

THUẬT GIẢI DIJKSTRA



THUẬT GIẢI BELLMAN-FORD

- **Bài toán:** Tìm các đường đi ngắn nhất từ một đỉnh s đến mọi đỉnh khác trong một đồ thị có trọng số là các số thực (có thể là số âm)

THUẬT GIẢI BELLMAN-FORD

Ý tưởng

- Ký hiệu d[v] là một cận trên của d(s,v), thuật giải kiểm tra và giảm d[v] theo tất cả các cạnh (u, v)
  - Nếu d[v]>d[u]+w(u,v) thì làm tốt cận trên d[v] bằng cách gán d[v]= d[u]+w(u,v)
  - Kết thúc quá trình giảm d[v], thuật giải kiểm tra tất cả các cạnh (u, v) nếu có d[v] > d[u] + w(u, v) thì xác định tồn tại chu trình âm, ngược lại các d[v] là độ dài đường đi ngắn nhất từ s đến v

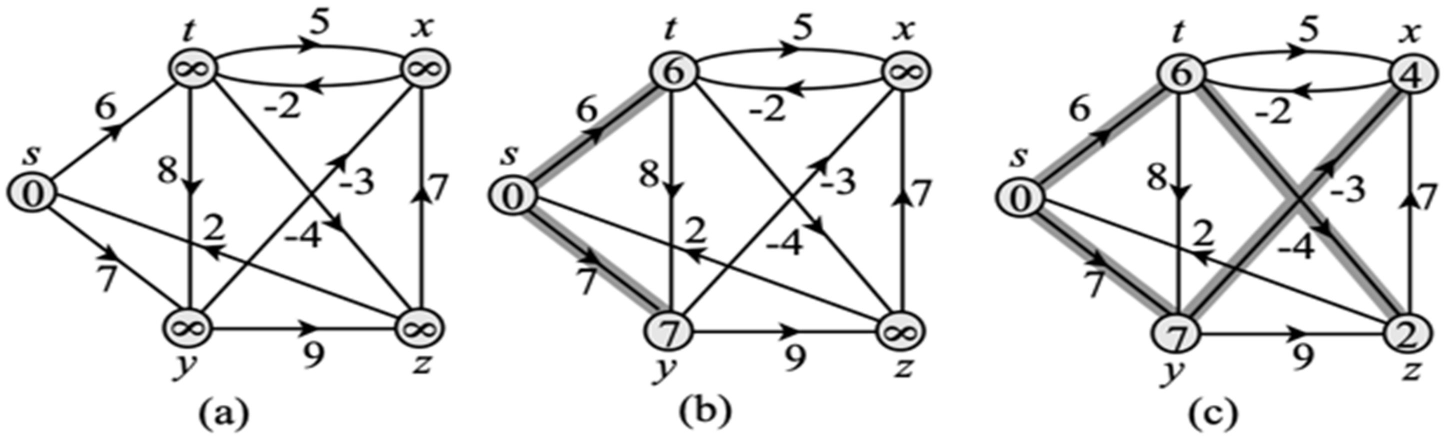
THUẬT GIẢI BELLMAN-FORD

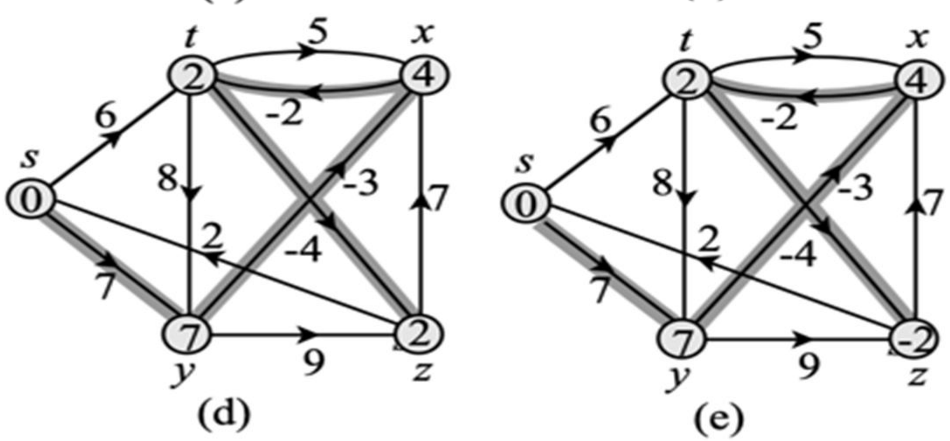
```
BELLMAN-FORD(G, w, s)
    INITIALIZESINGLE-SOURCE(G, s)
    for i =1 to |V[G]|−1
        for each (u, v) ∈ E[G]
            RELAX(u, v, w)
    for each (u, v) ∈ E[G]
        if d[v] > d[u]  + w(u, v)
            return FALSE
    return TRUE
```

THUẬT GIẢI BELLMAN-FORD

- Thời gian khởi tạo ở dòng lệnh 1 là $\Theta(V)$
- Mỗi chuyển trong |V|-1 chuyến trên các cạnh thực thi các dòng lệnh 2-4 chi phí là $\Theta(E)$
- Chi phí của vòng lặp **for** gồm các dòng lệnh 5-7 là $O(E)$
- Thời gian chạy của Thuật toán Bellman-Ford là $O(VE)$

THUẬT GIẢI BELLMAN-FORD





Các cạnh được Relax theo thứ tự (t, x), (t, y), (t, z), (x, t), (y, x), (y, z), (z, x), (z, s), (s, t), (s, y) bốn lượt

ĐỌC VÀ TÌM HIỂU Ở NHÀ

- Đọc chương 24 sách Introduction to Algorithms của Cormen và cộng sự
- Làm bài tập về nhà chương 4 đã cho trong DS bài tập