

Shortest Path

Single Source Shortest Path

Rajesh Kumar Tripathi,
Assistant Professor, Dept. CEA,
GLA University, Mathura

Relaxation

Algorithms keep track of $d[v]$, $\pi[v]$. **Initialized** as follows:

```

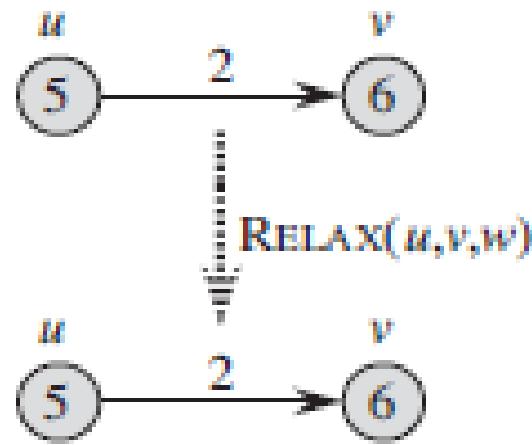
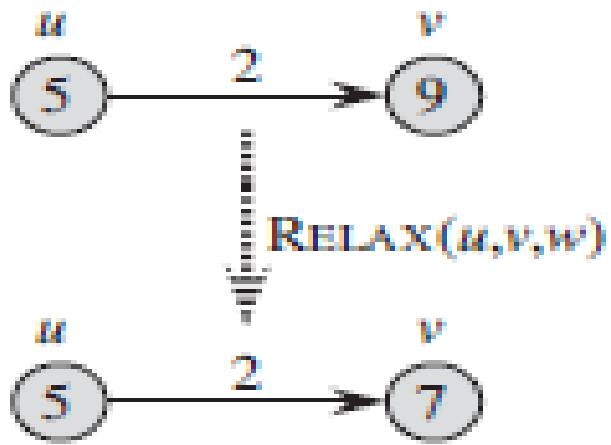
Initialize(G, s)
    for each v ∈ V[G] do
        d[v] := ∞;
        π[v] := NIL
    end;
    d[s] := 0
  
```

These values are changed when an edge (u, v) is **relaxed**:

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Relaxation



Dijkstra's Algorithm

Assumes **no negative-weight edges**.

Maintains a set **S** of vertices whose SP from s has been determined.

Repeatedly selects u in V–S with minimum SP estimate (**greedy choice**)

Store V–S in **priority queue Q**.

```

Initialize(G, s)
  for each v ∈ V[G] do
    d[v] := ∞;
    π[v] := NIL
  end;
  d[s] := 0

```

```

Relax(u, v, w)
  if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
    π[v] := u
  end

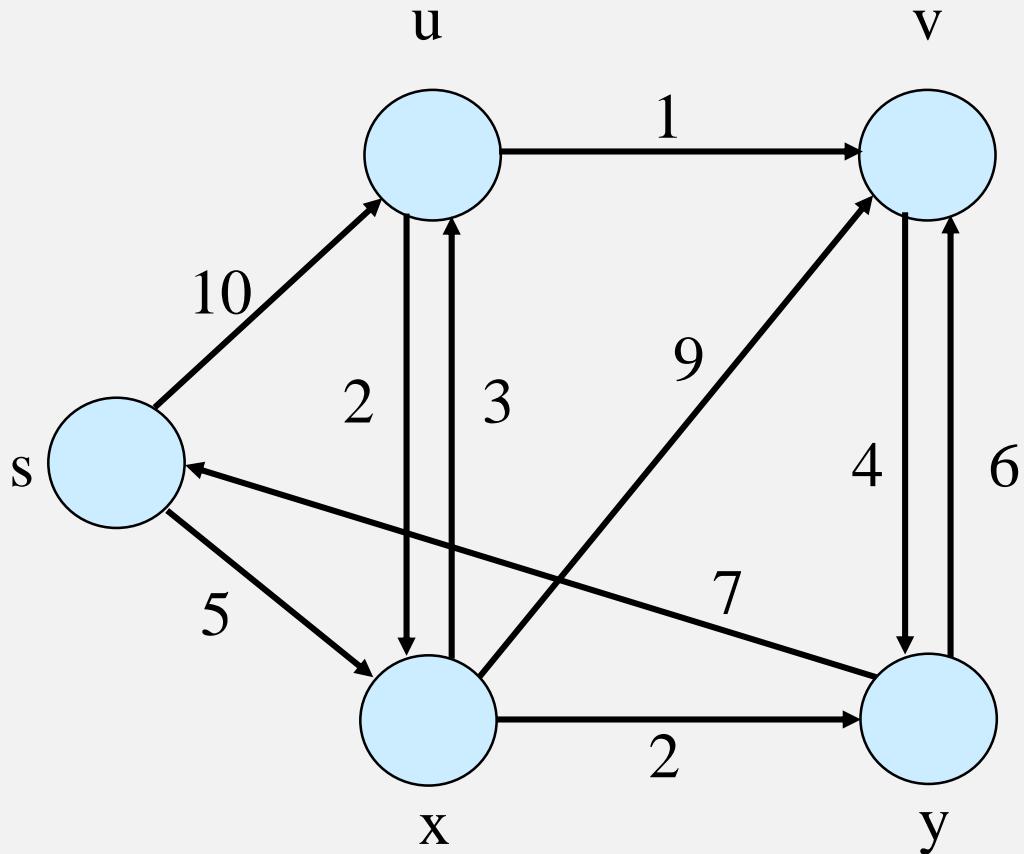
```

```

Initialize(G, s);
S := ∅;
Q := V[G];
while Q ≠ ∅ do
  u := Extract-Min(Q);
  S := S ∪ {u};
  for each v ∈ Adj[u] do
    Relax(u, v, w)
  end
end

```

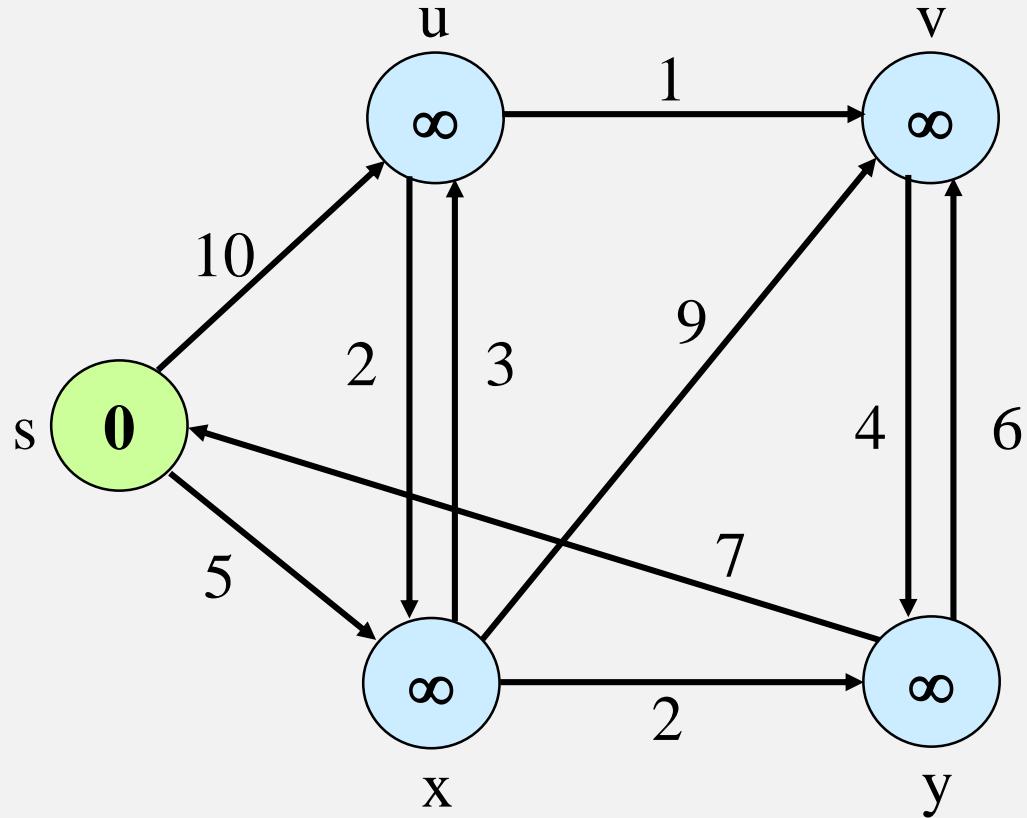
Example



```

Initialize( $G, s$ )
  for each  $v \in V[G]$  do
     $d[v] := \infty$ ;
     $\pi[v] := \text{NIL}$ 
  end;
   $d[s] := 0$ 
  
```

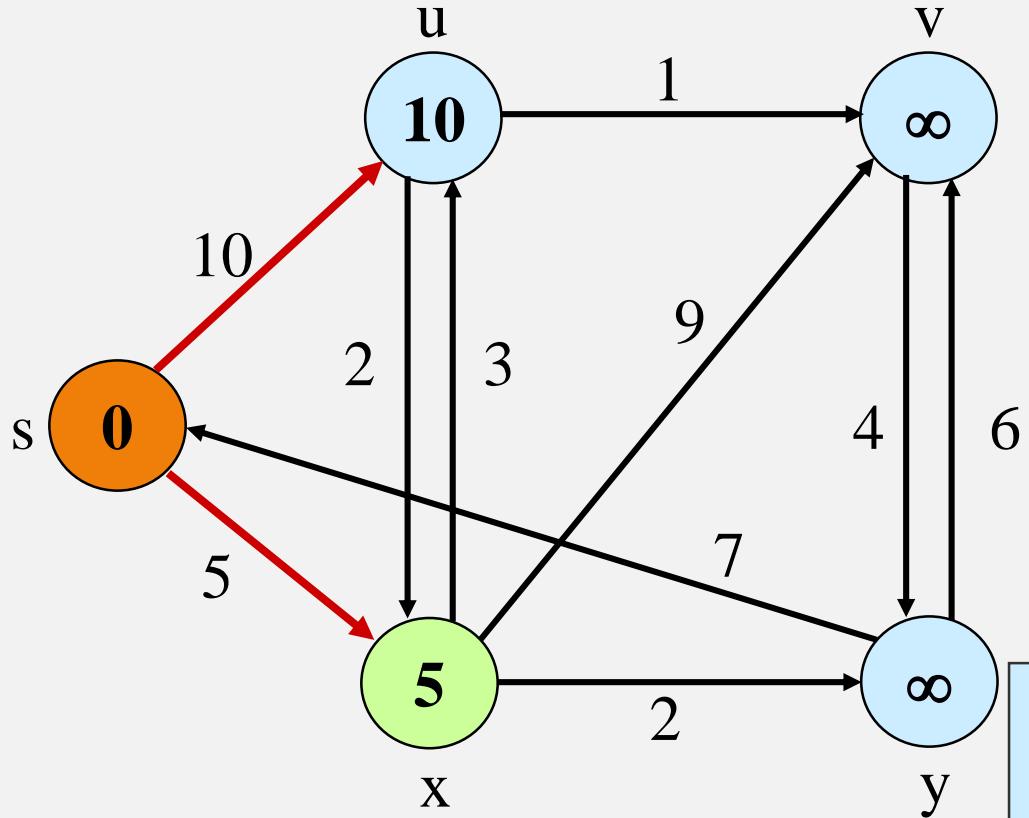
Example



```

Initialize(G, s)
  for each  $v \in V[G]$  do
     $d[v] := \infty;$ 
     $\pi[v] := \text{NIL}$ 
  end;
   $d[s] := 0$ 
  
```

Example



```

Initialize( $G, s$ );
 $S := \emptyset$ ;
 $Q := V[G]$ ;
while  $Q \neq \emptyset$  do
     $u := \text{Extract-Min}(Q)$ ;
     $S := S \cup \{u\}$ ;
    for each  $v \in \text{Adj}[u]$  do
        Relax( $u, v, w$ )
    end
end

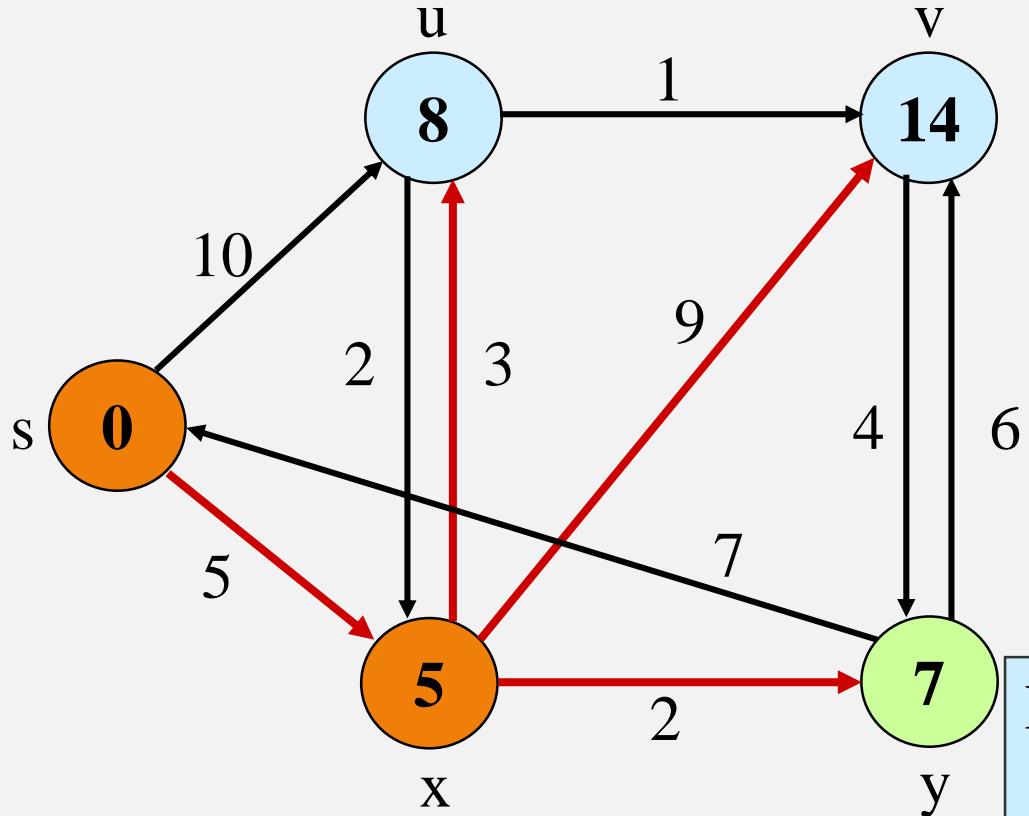
```

```

Relax( $u, v, w$ )
if  $d[v] > d[u] + w(u, v)$  then
     $d[v] := d[u] + w(u, v)$ ;
     $\pi[v] := u$ 
end

```

Example



```

Initialize(G, s);
S := ∅;
Q := V[G];
while Q ≠ ∅ do
    u := Extract-Min(Q);
    S := S ∪ {u};
    for each v ∈ Adj[u] do
        Relax(u, v, w)
    end
end

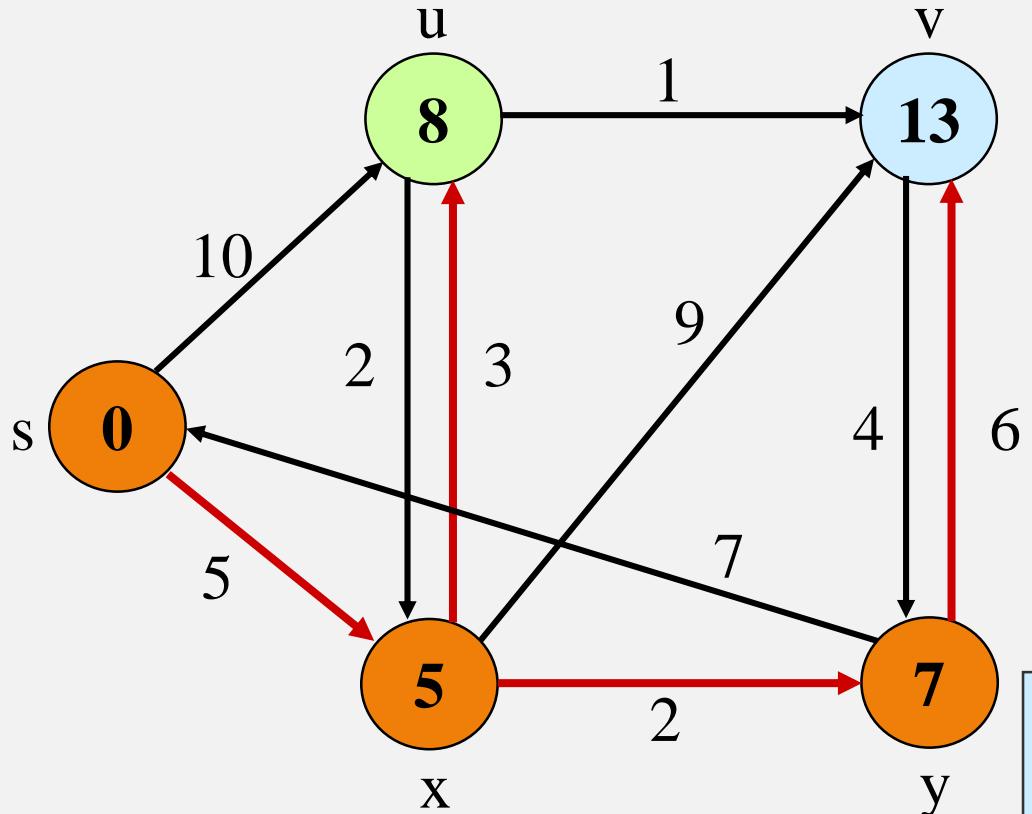
```

```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
    π[v] := u
end

```

Example



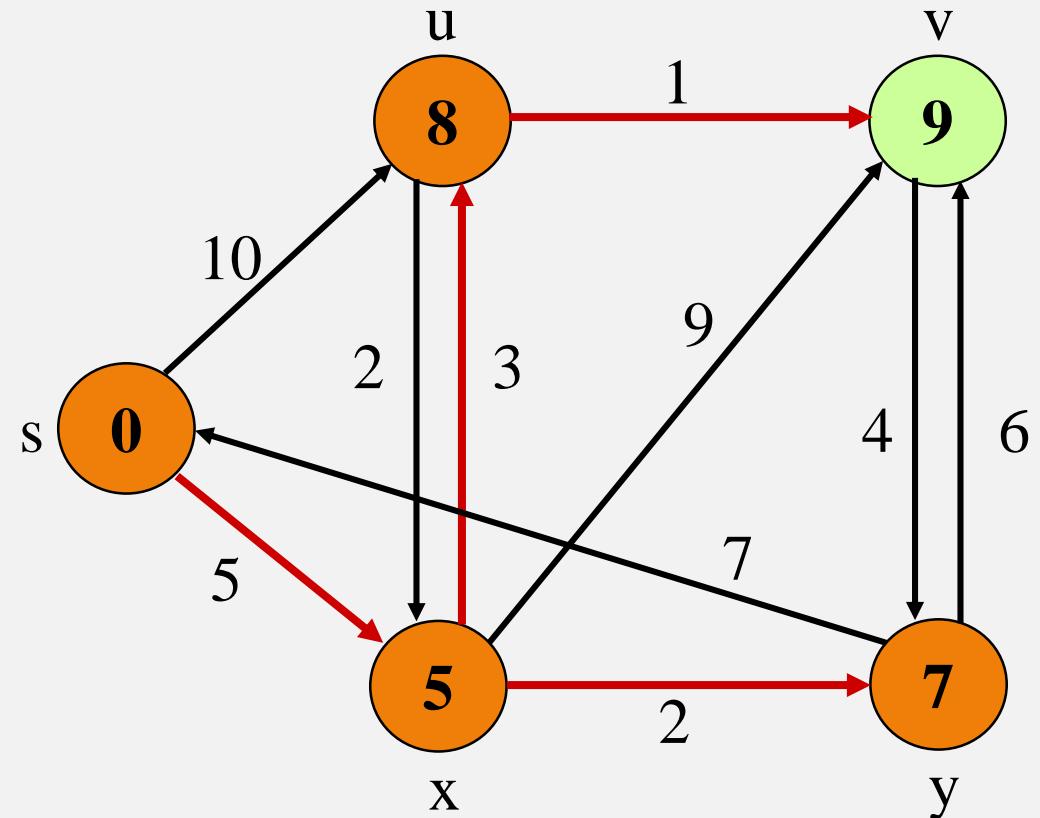
```

Initialize(G, s);
S := ∅;
Q := V[G];
while Q ≠ ∅ do
  u := Extract-Min(Q);
  S := S ∪ {u};
  for each v ∈ Adj[u] do
    Relax(u, v, w)
  end
end
  
```

```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
  d[v] := d[u] + w(u, v);
  π[v] := u
end
  
```

Example



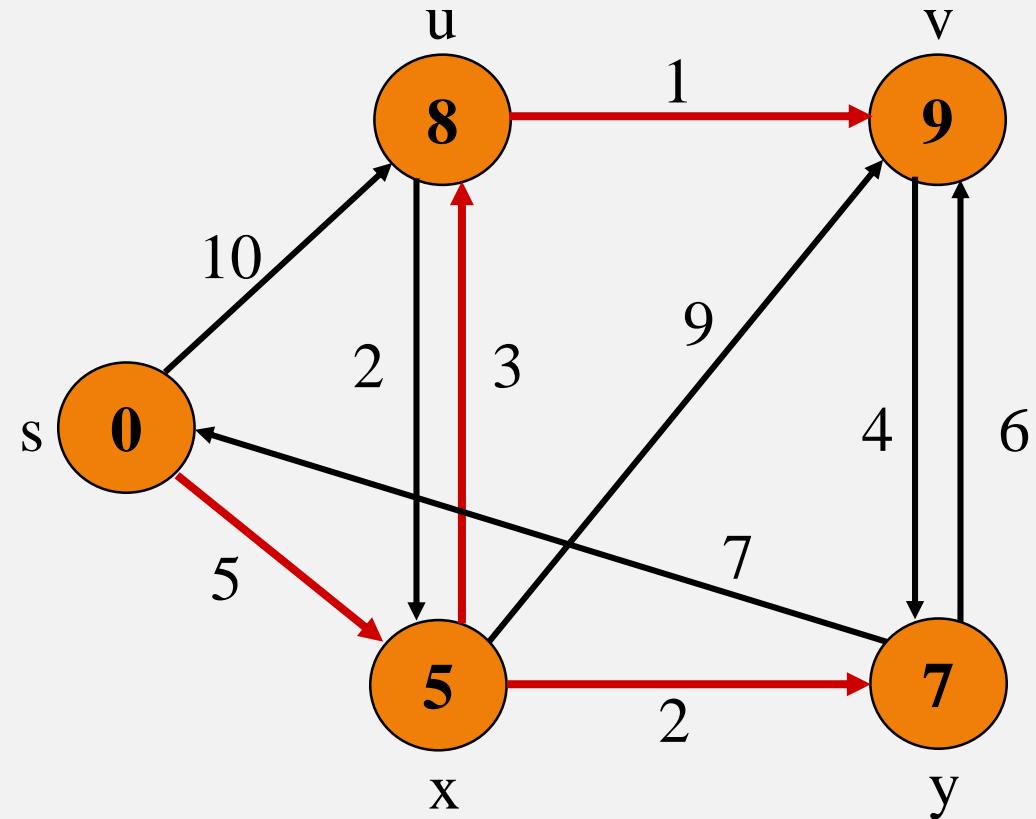
```

Initialize(G, s);
S := ∅;
Q := V[G];
while Q ≠ ∅ do
  u := Extract-Min(Q);
  S := S ∪ {u};
  for each v ∈ Adj[u] do
    Relax(u, v, w)
  end
end
  
```

```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
  d[v] := d[u] + w(u, v);
  π[v] := u
end
  
```

Example



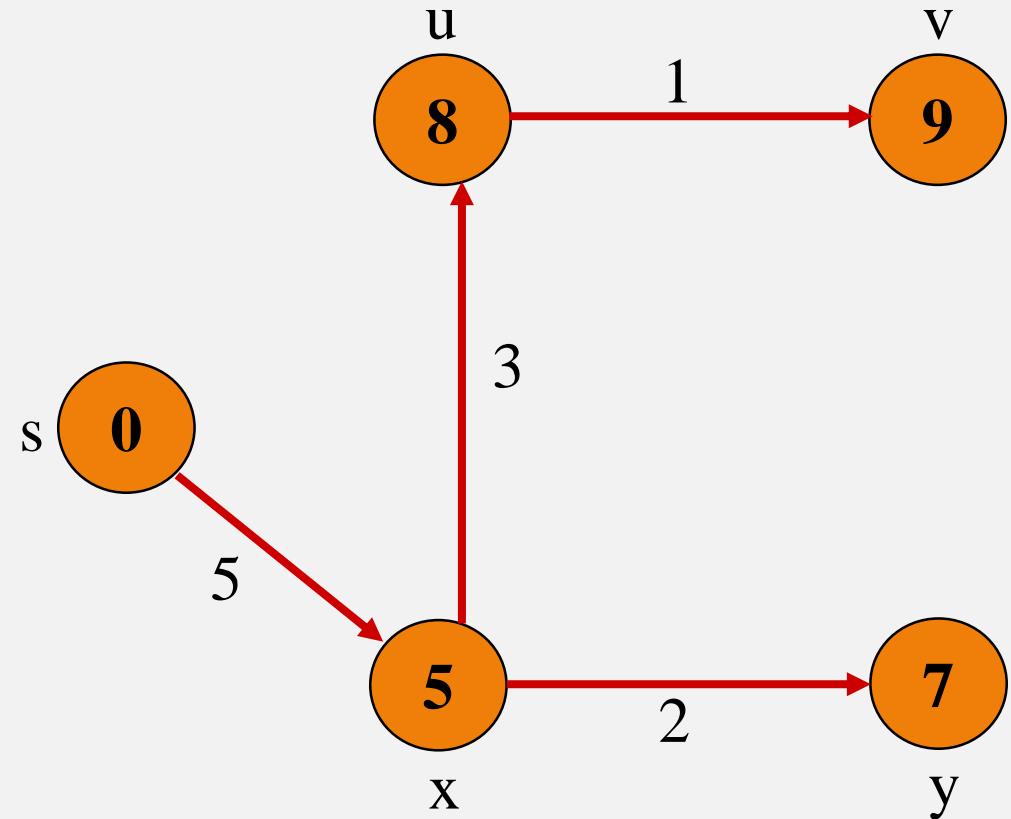
```

Initialize(G, s);
S := ∅;
Q := V[G];
while Q ≠ ∅ do
  u := Extract-Min(Q);
  S := S ∪ {u};
  for each v ∈ Adj[u] do
    Relax(u, v, w)
  end
end
  
```

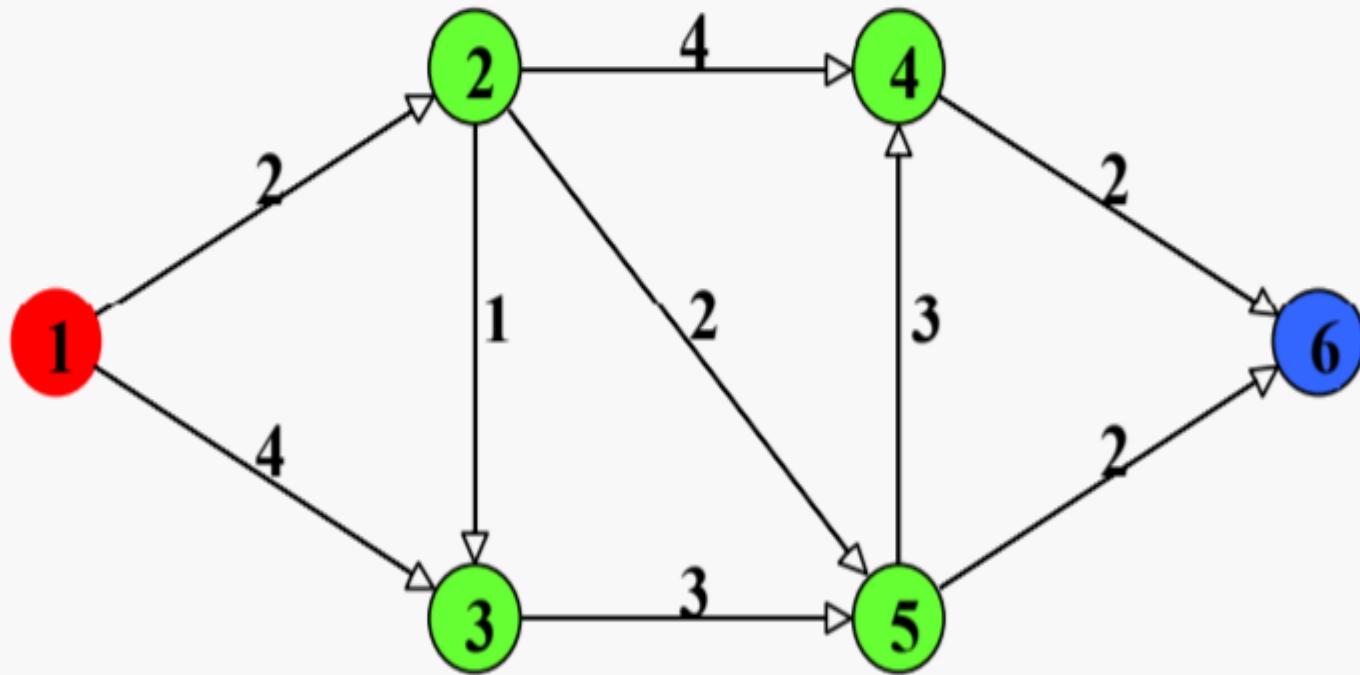
```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
  d[v] := d[u] + w(u, v);
  π[v] := u
end
  
```

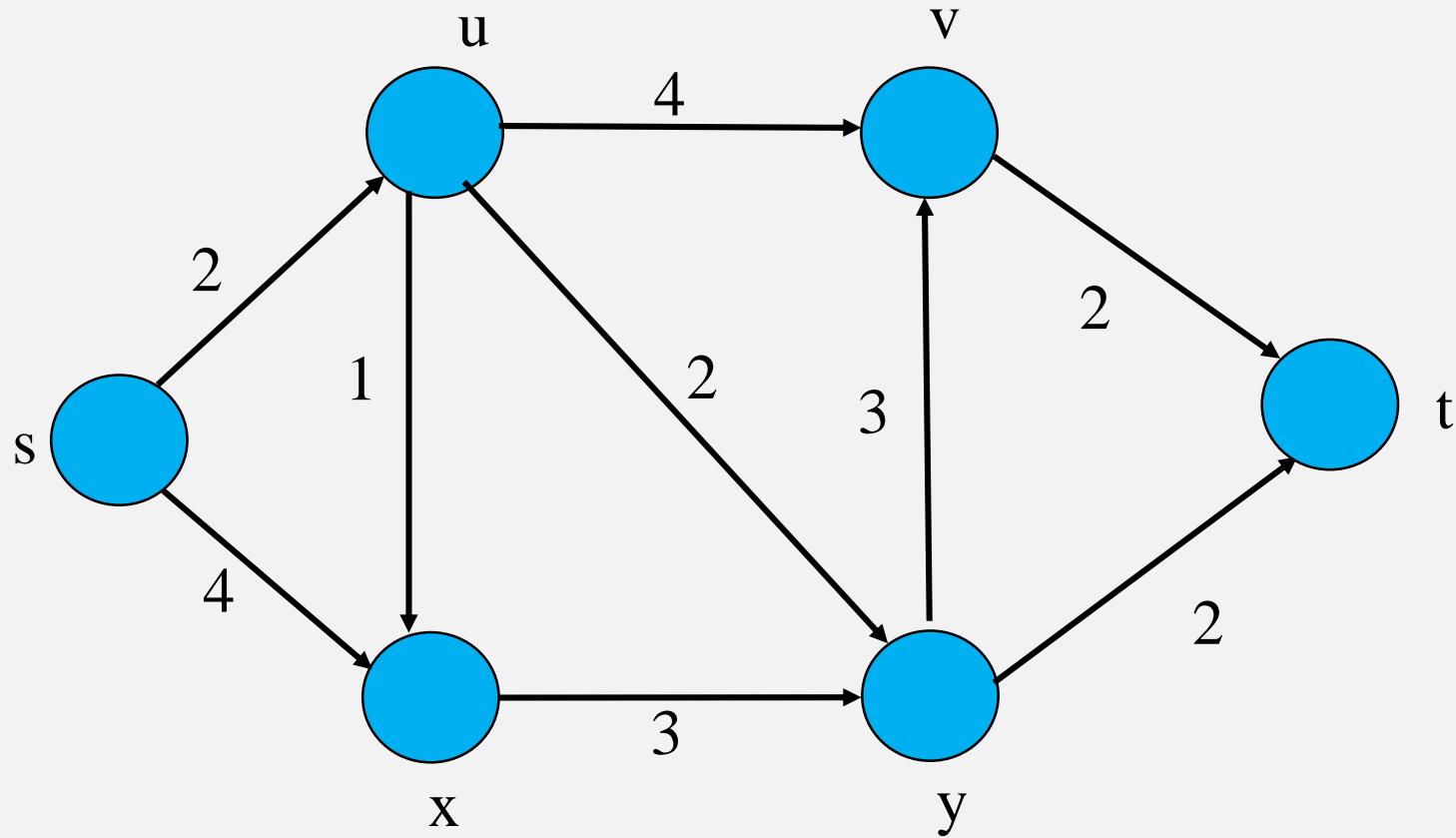
Example



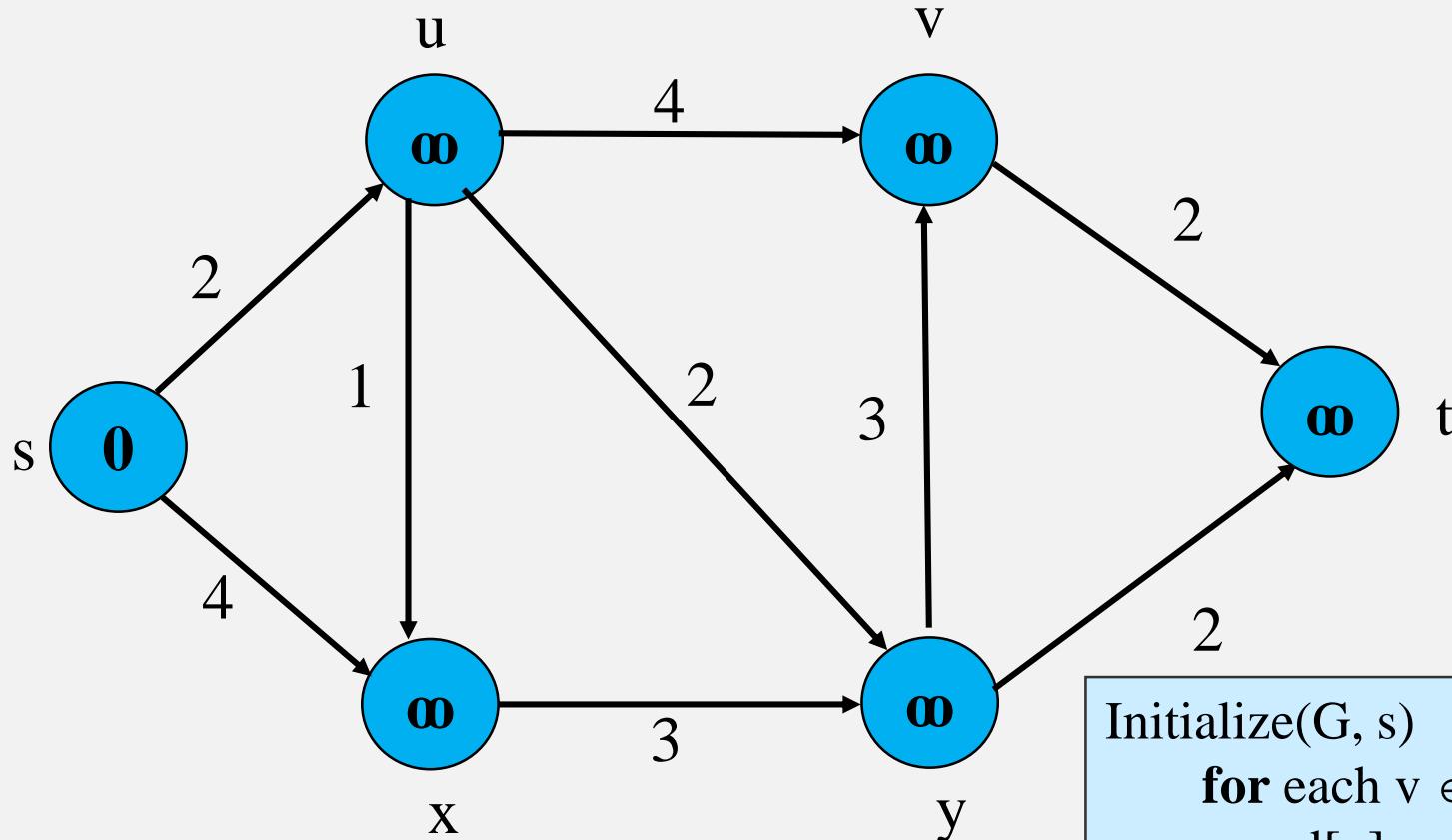
Example



Example



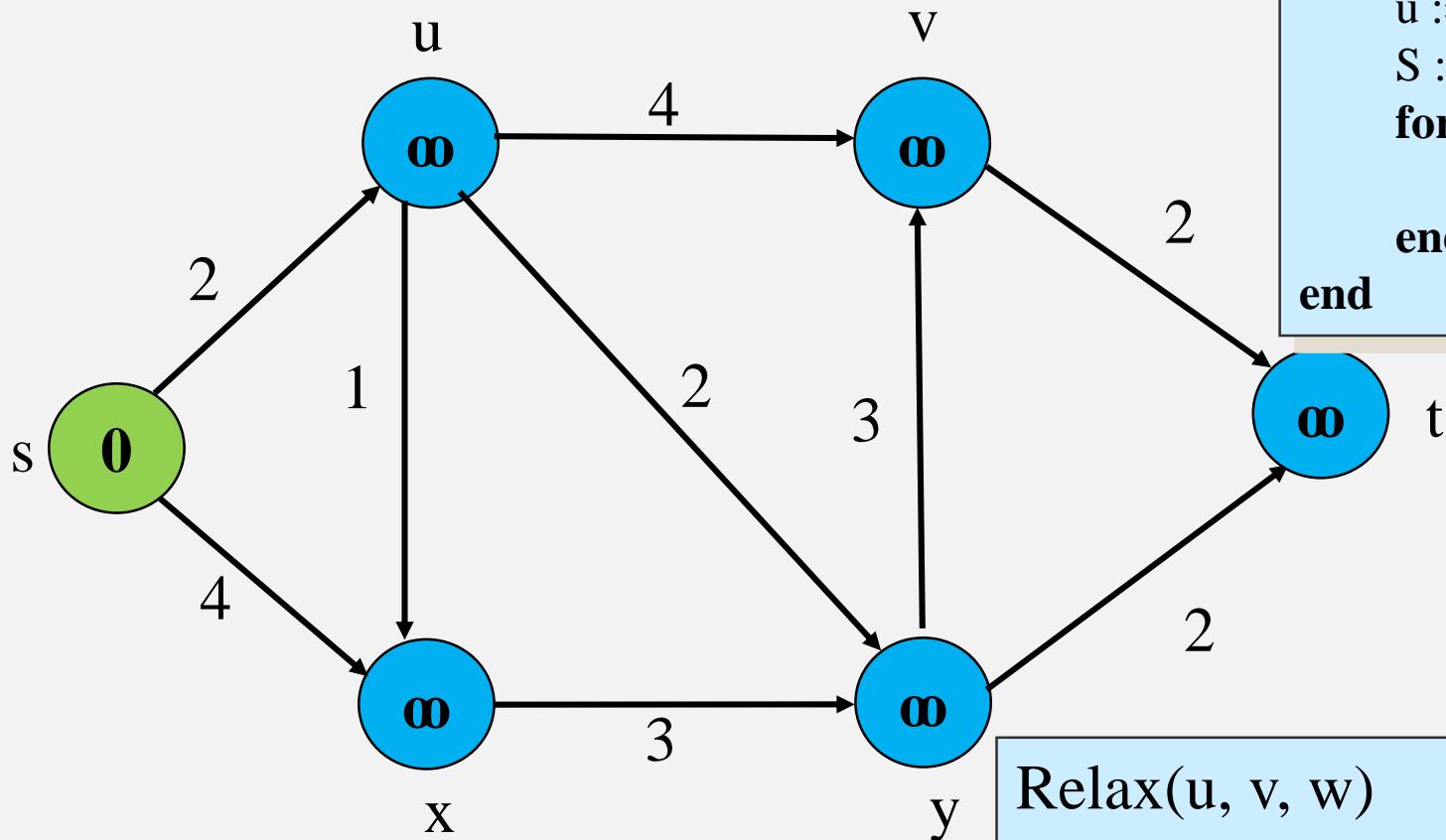
Example



```

Initialize( $G, s$ )
  for each  $v \in V[G]$  do
     $d[v] := \infty;$ 
     $\pi[v] := \text{NIL}$ 
  end;
   $d[s] := 0$ 
  
```

Example



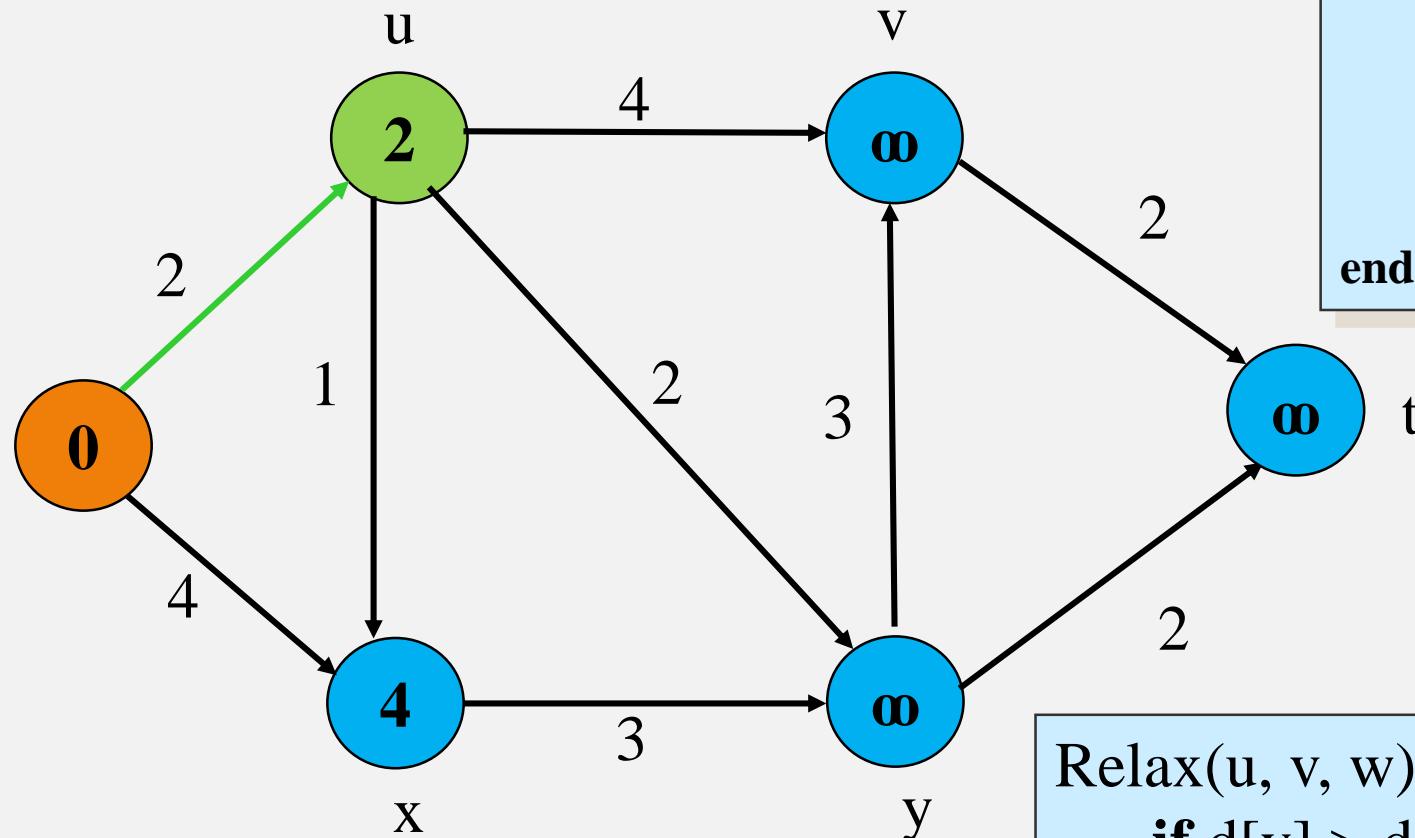
```

Initialize(G, s);
S := ∅;
Q := V[G];
while Q ≠ ∅ do
    u := Extract-Min(Q);
    S := S ∪ {u};
    for each v ∈ Adj[u] do
        Relax(u, v, w)
    end
end
  
```

```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
    π[v] := u
end
  
```

Example



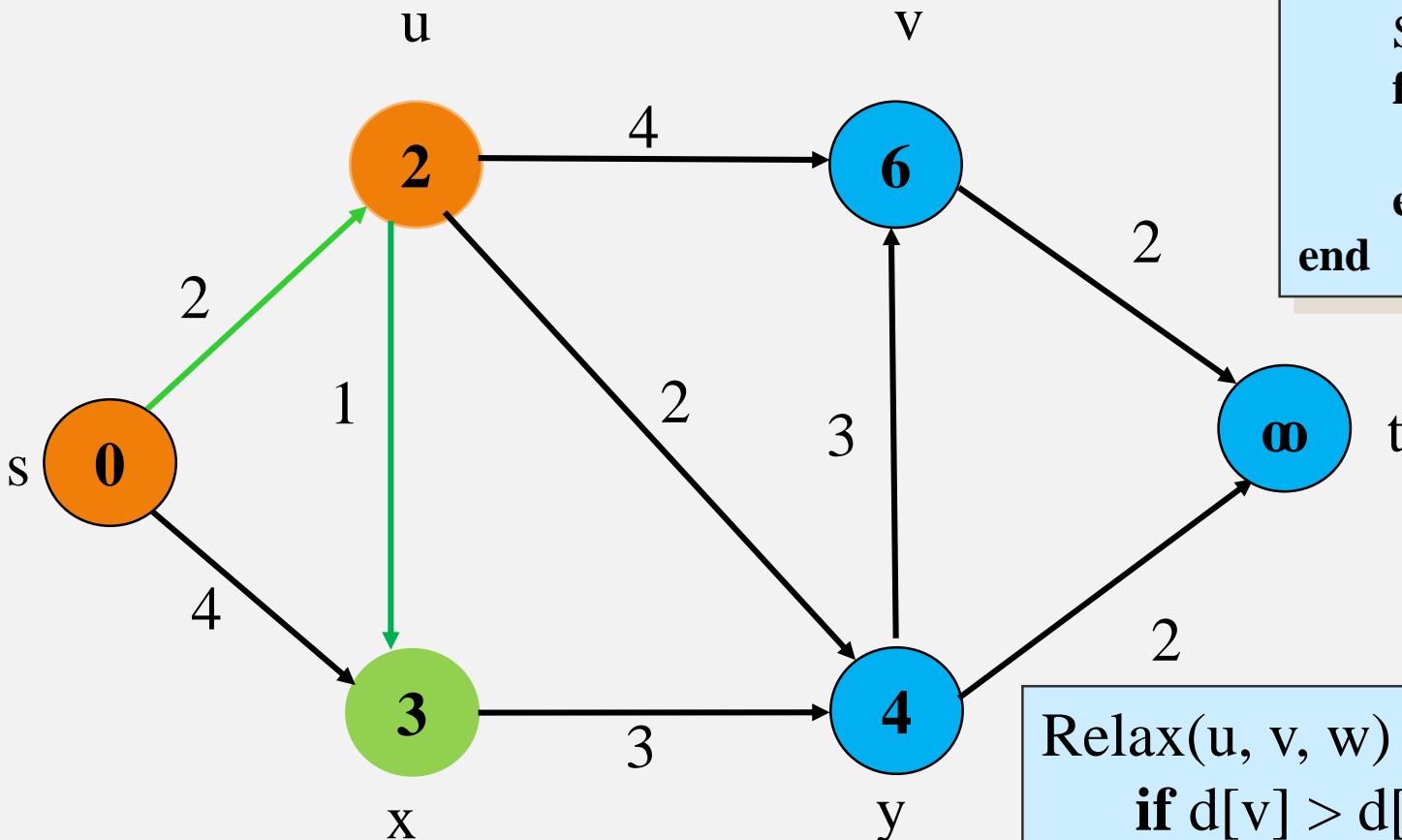
```

Initialize(G, s);
S := ∅;
Q := V[G];
while Q ≠ ∅ do
    u := Extract-Min(Q);
    S := S ∪ {u};
    for each v ∈ Adj[u] do
        Relax(u, v, w)
    end
end
  
```

```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
    π[v] := u
end
  
```

Example



```

Initialize(G, s);
S := ∅;
Q := V[G];
while Q ≠ ∅ do
    u := Extract-Min(Q);
    S := S ∪ {u};
    for each v ∈ Adj[u] do
        Relax(u, v, w)
    end
end

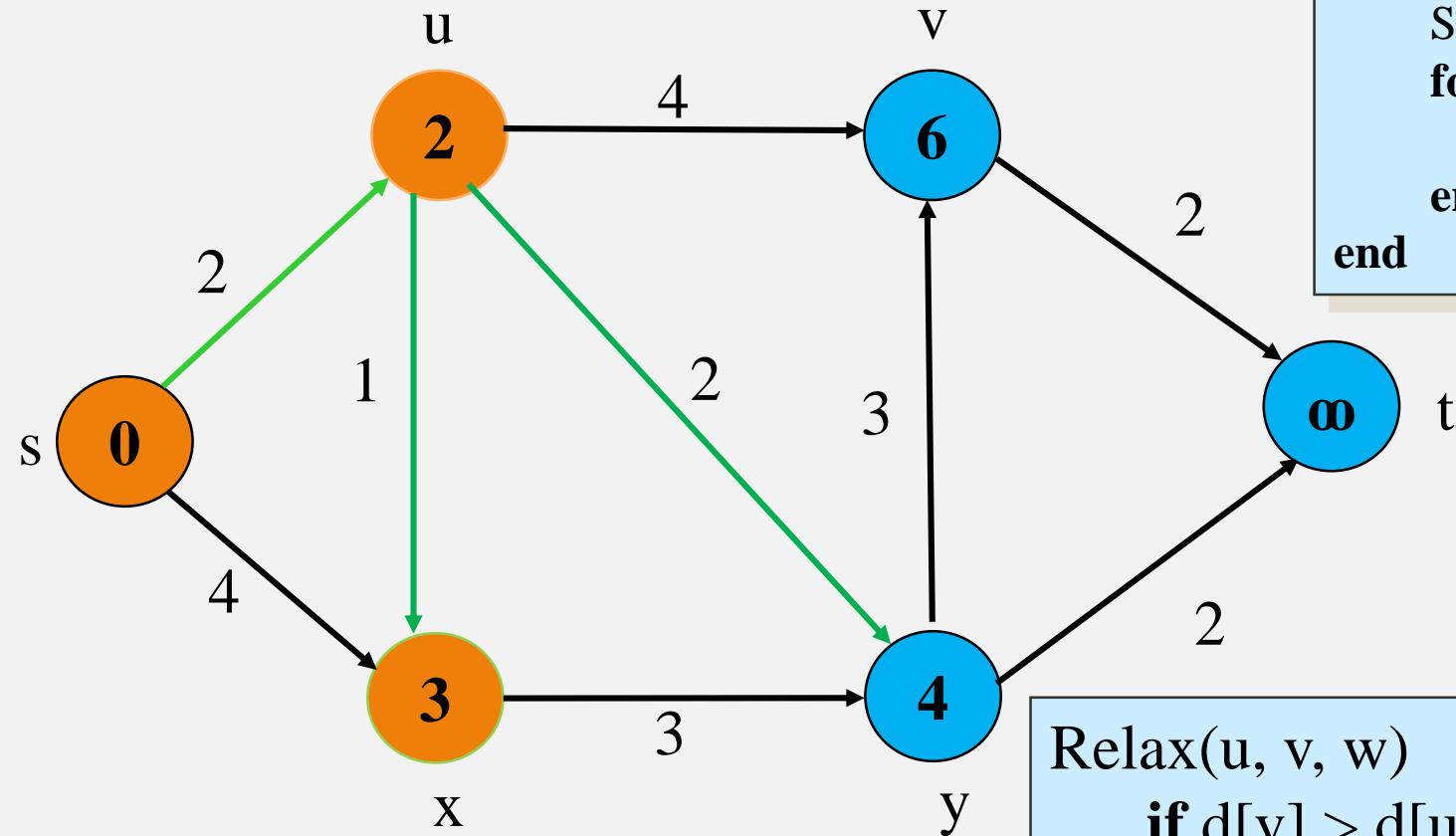
```

```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
    π[v] := u
end

```

Example



```

Initialize(G, s);
S := ∅;
Q := V[G];
while Q ≠ ∅ do
    u := Extract-Min(Q);
    S := S ∪ {u};
    for each v ∈ Adj[u] do
        Relax(u, v, w)
    end
end

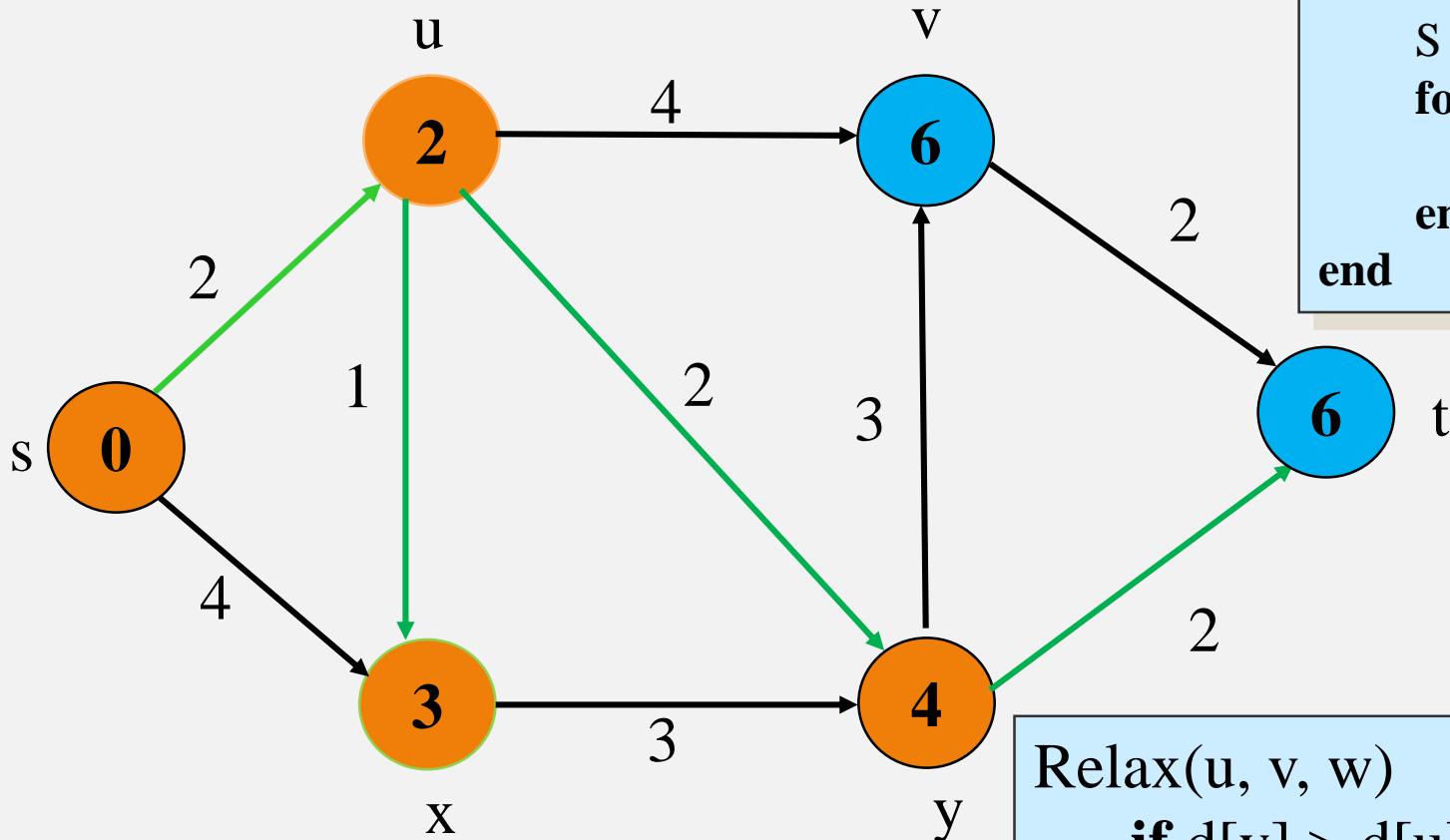
```

```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
    π[v] := u
end

```

Example



```

Initialize(G, s);
S := ∅;
Q := V[G];
while Q ≠ ∅ do
    u := Extract-Min(Q);
    S := S ∪ {u};
    for each v ∈ Adj[u] do
        Relax(u, v, w)
    end
end

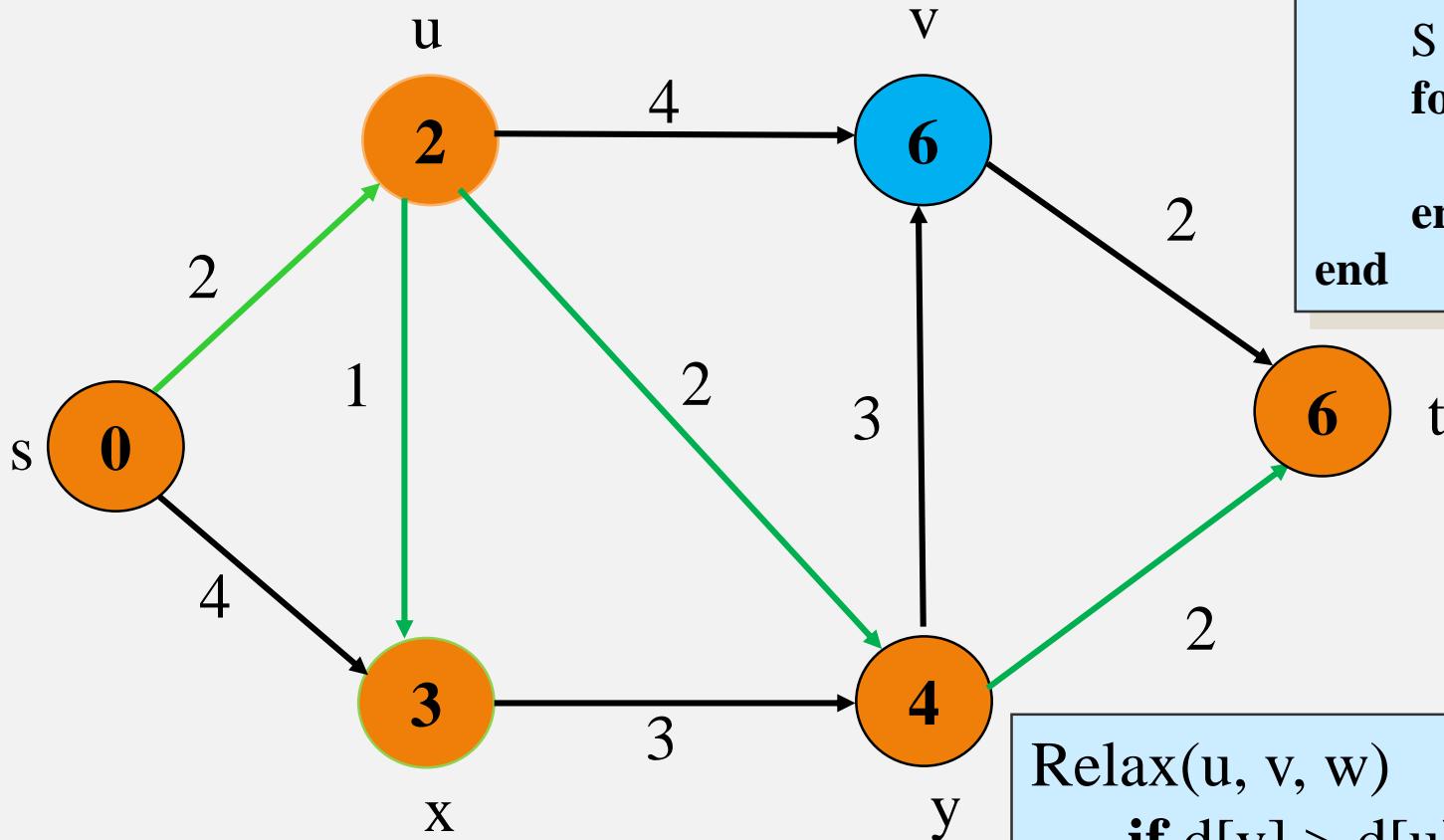
```

```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
    π[v] := u
end

```

Example



```

Initialize(G, s);
S := ∅;
Q := V[G];
while Q ≠ ∅ do
    u := Extract-Min(Q);
    S := S ∪ {u};
    for each v ∈ Adj[u] do
        Relax(u, v, w)
    end
end

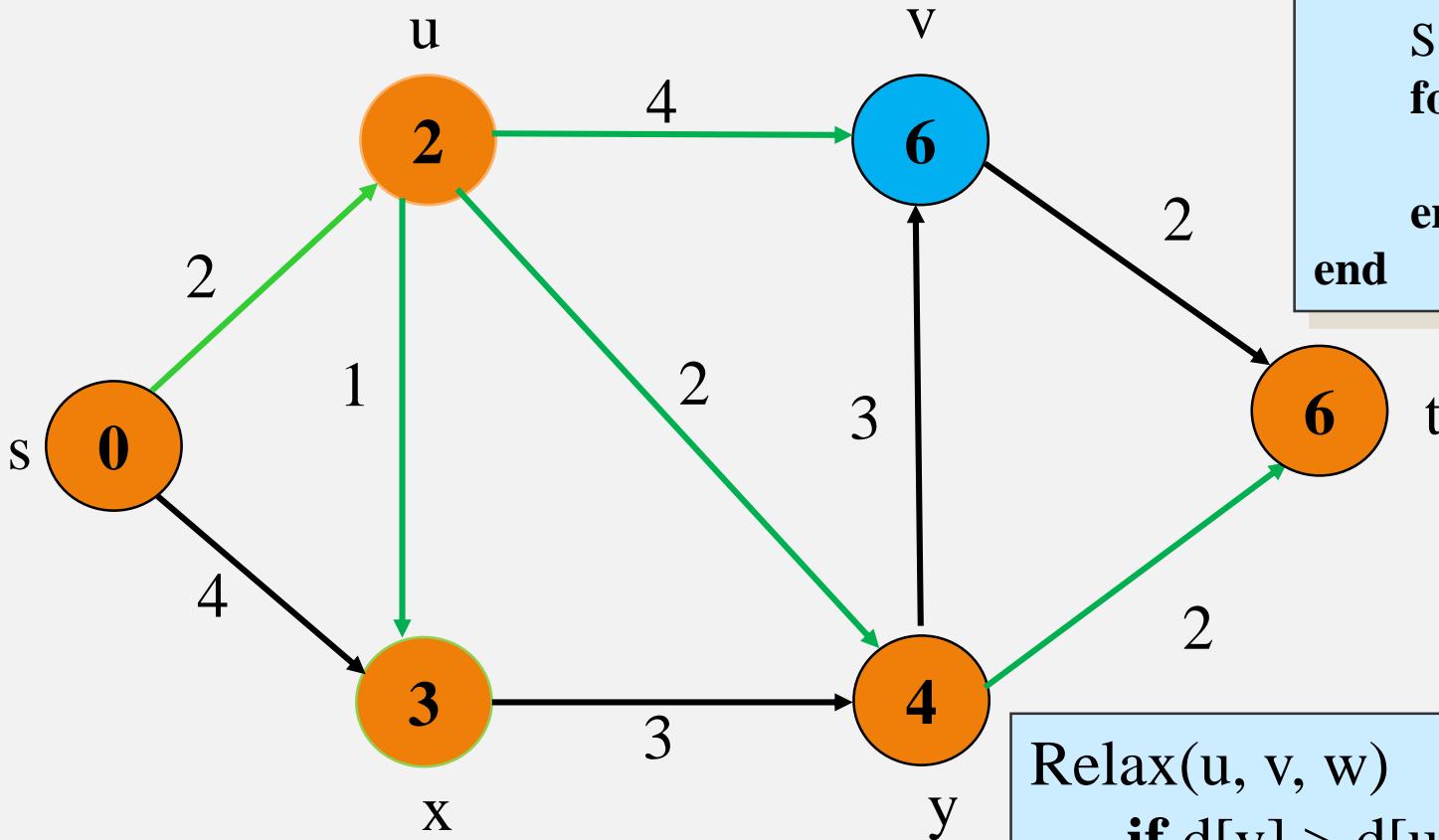
```

```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
    π[v] := u
end

```

Example



```

Initialize(G, s);
S := ∅;
Q := V[G];
while Q ≠ ∅ do
    u := Extract-Min(Q);
    S := S ∪ {u};
    for each v ∈ Adj[u] do
        Relax(u, v, w)
    end
end

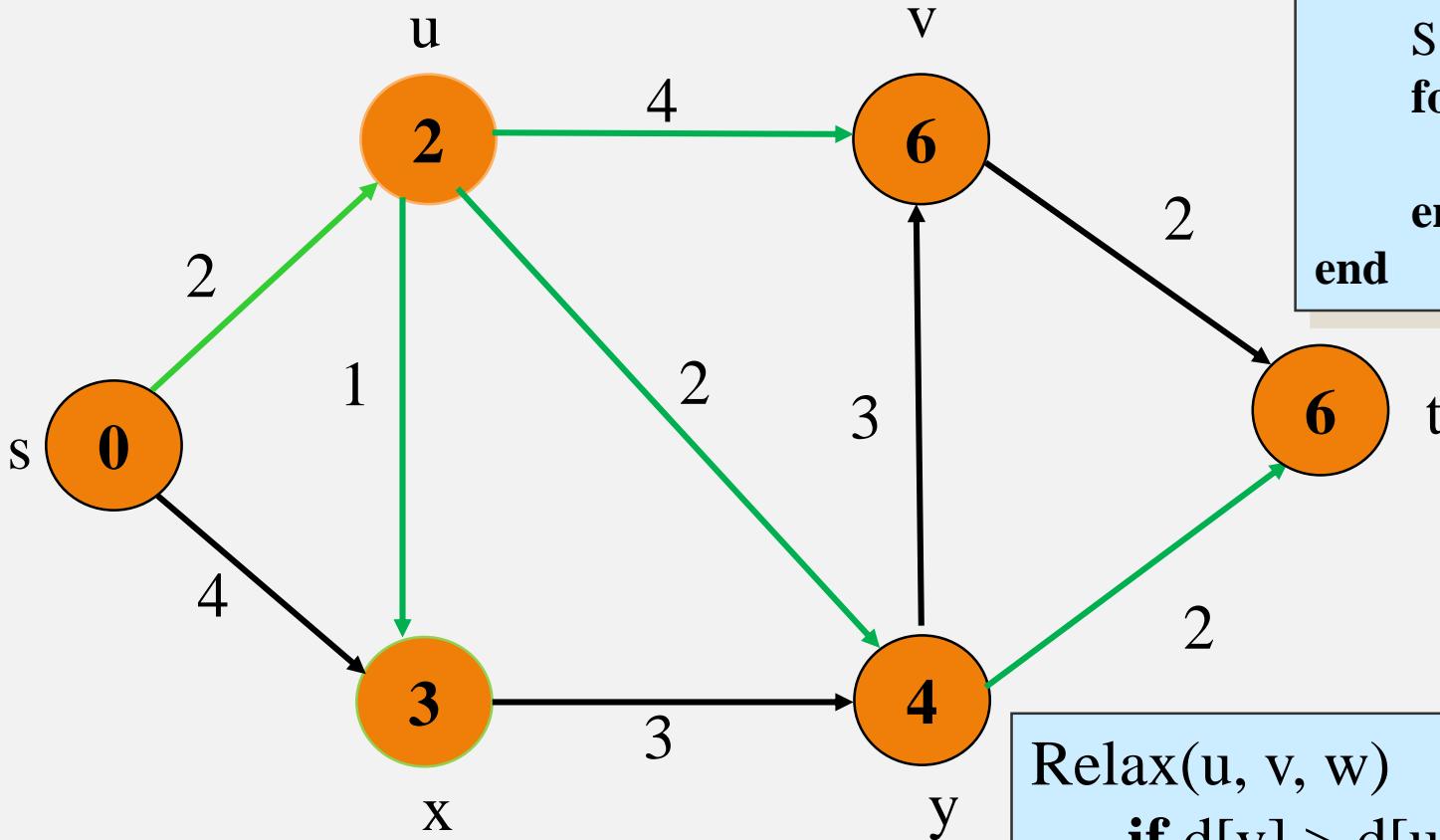
```

```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
    π[v] := u
end

```

Example



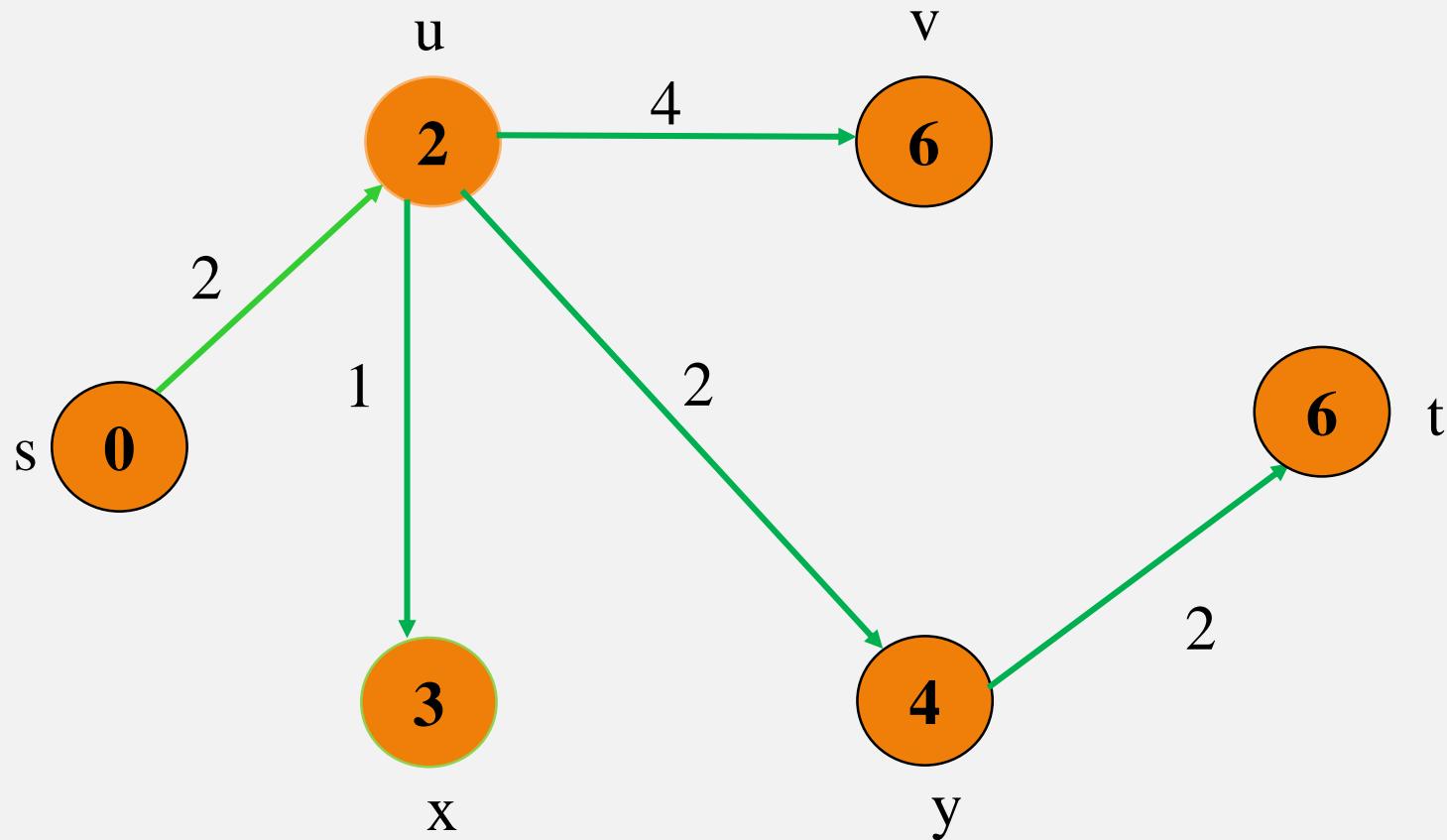
```

Initialize(G, s);
S := ∅;
Q := V[G];
while Q ≠ ∅ do
  u := Extract-Min(Q);
  S := S ∪ {u};
  for each v ∈ Adj[u] do
    Relax(u, v, w)
  end
end
  
```

```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
  d[v] := d[u] + w(u, v);
  π[v] := u
end
  
```

Example



Thank You



Bellman-Ford Algorithm

Initialize(G, s)

```

for each  $v \in V[G]$  do
     $d[v] := \infty;$ 
     $\pi[v] := \text{NIL}$ 
end;
 $d[s] := 0$ 
```

Relax(u, v, w)

```

if  $d[v] > d[u] + w(u, v)$  then
     $d[v] := d[u] + w(u, v);$ 
     $\pi[v] := u$ 
end
```

Bellman-Ford Algorithm

Can have negative-weight edges.

Will “detect” reachable negative-weight cycles.

```

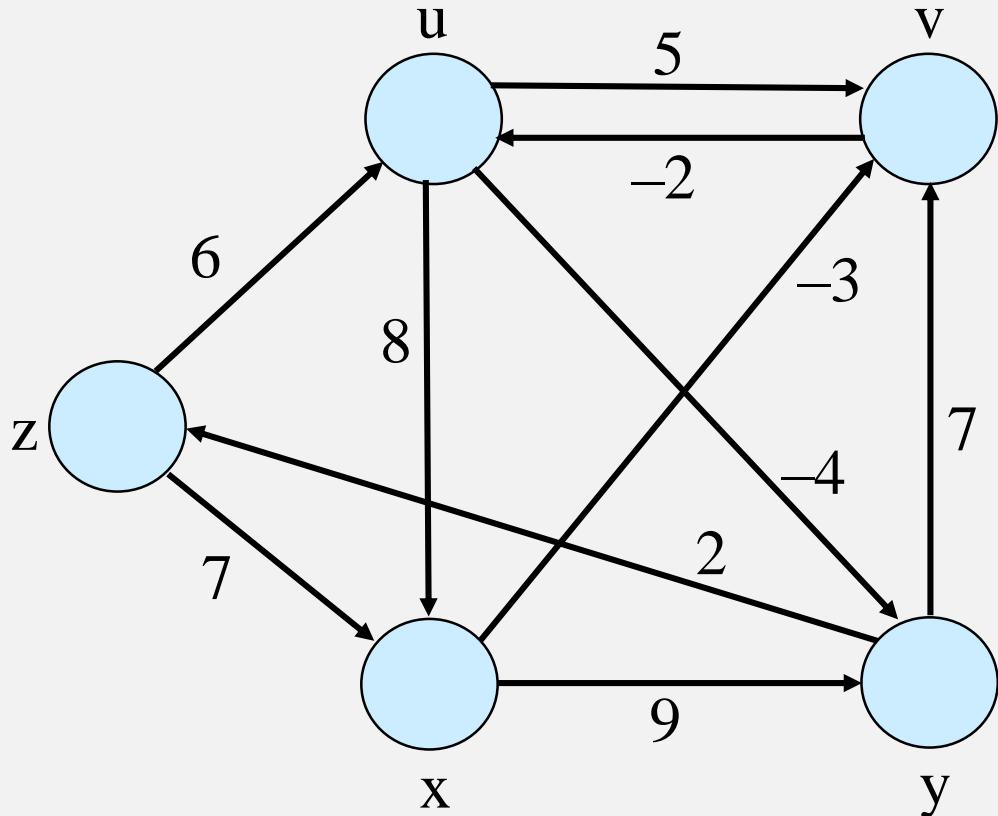
Initialize(G, s);
for i := 1 to |V[G]| − 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

```

Time Complexity
 $O(VE)$

- If Bellman-Ford has not converged after $V(G) - 1$ iterations, then there cannot be a shortest path tree, so there must be a negative weight cycle.

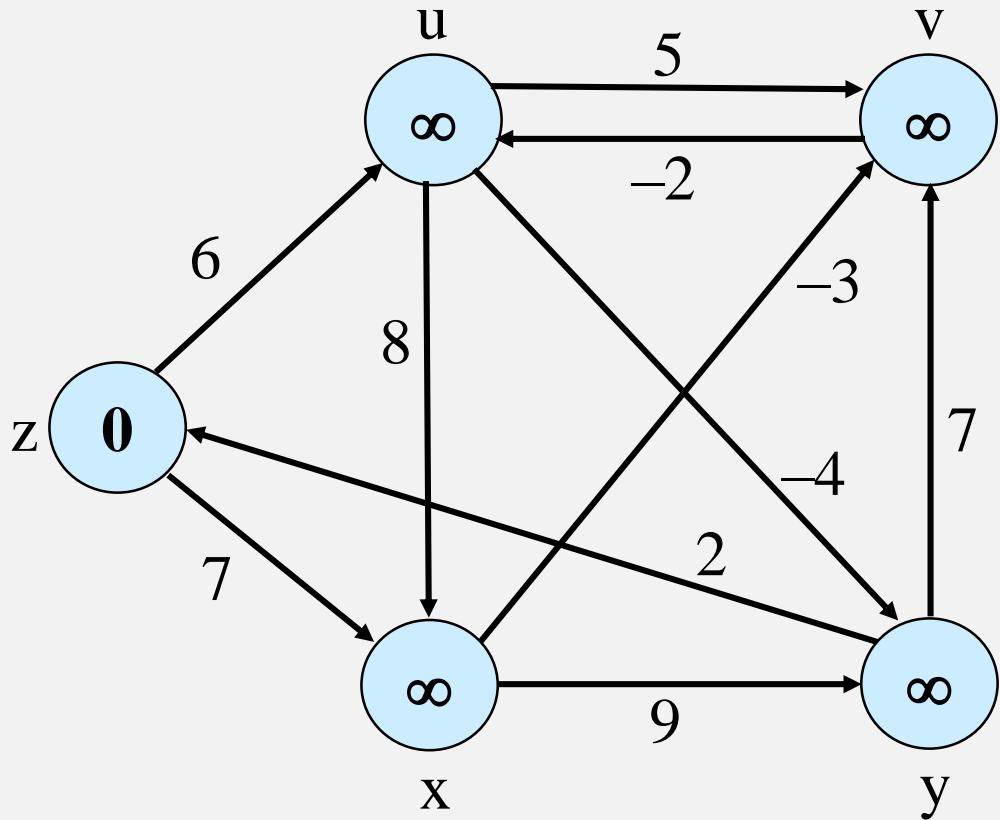
Example



```

Initialize(G, s)
  for each  $v \in V[G]$  do
     $d[v] := \infty;$ 
     $\pi[v] := \text{NIL}$ 
  end;
   $d[s] := 0$ 
  
```

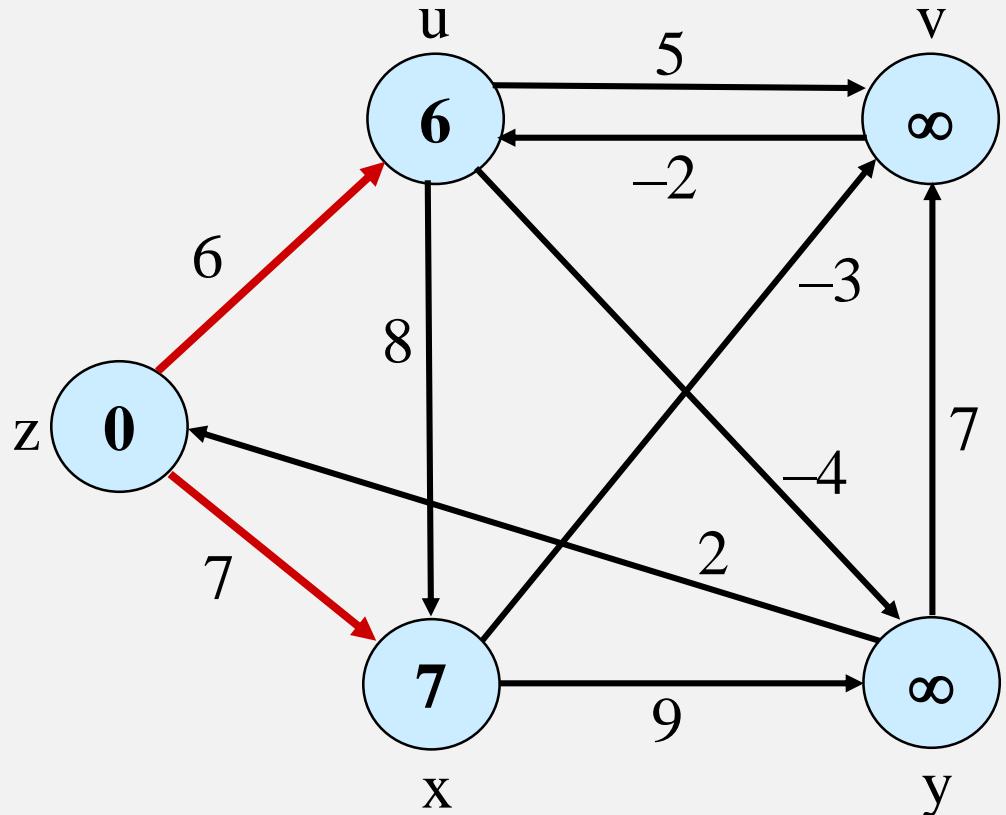
Example



```

Initialize( $G, s$ )
  for each  $v \in V[G]$  do
     $d[v] := \infty;$ 
     $\pi[v] := \text{NIL}$ 
  end;
   $d[s] := 0$ 
  
```

Example



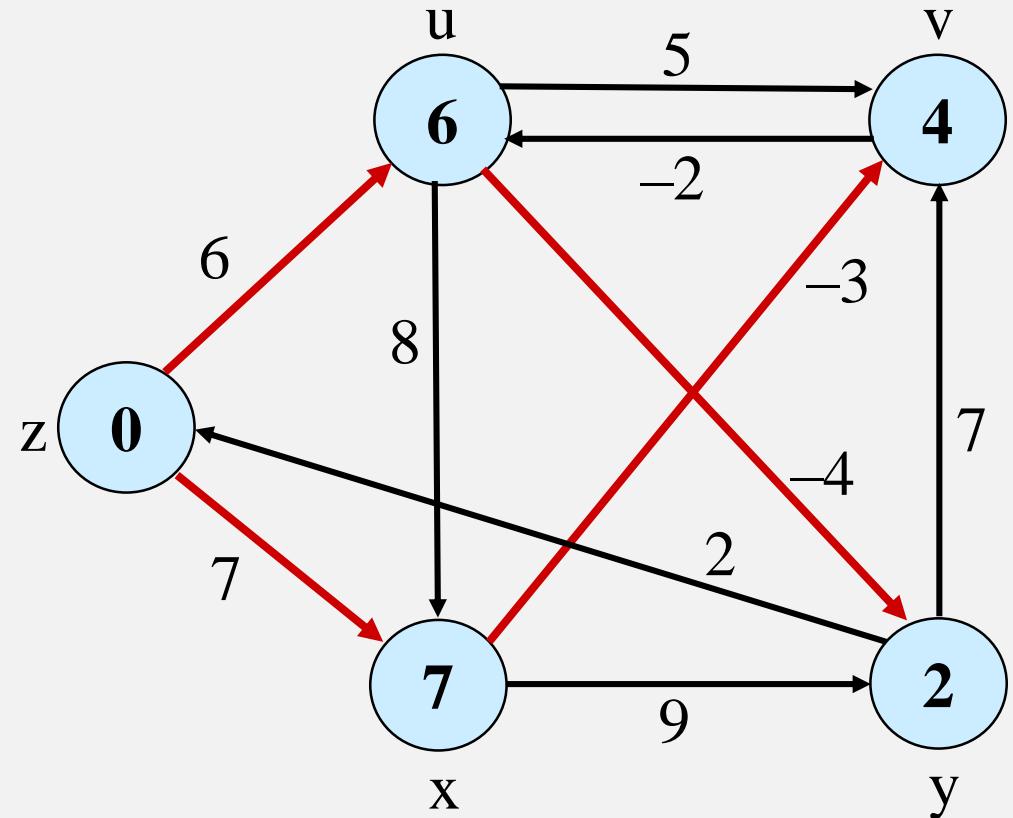
```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
  for each (u, v) in E[G] do
    Relax(u, v, w)
  end
end;
for each (u, v) in E[G] do
  if d[v] > d[u] + w(u, v) then
    return false
  end
end;
return true
  
```

```

Relax(u, v, w)
  if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
    π[v] := u
  end
  
```

Example



```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

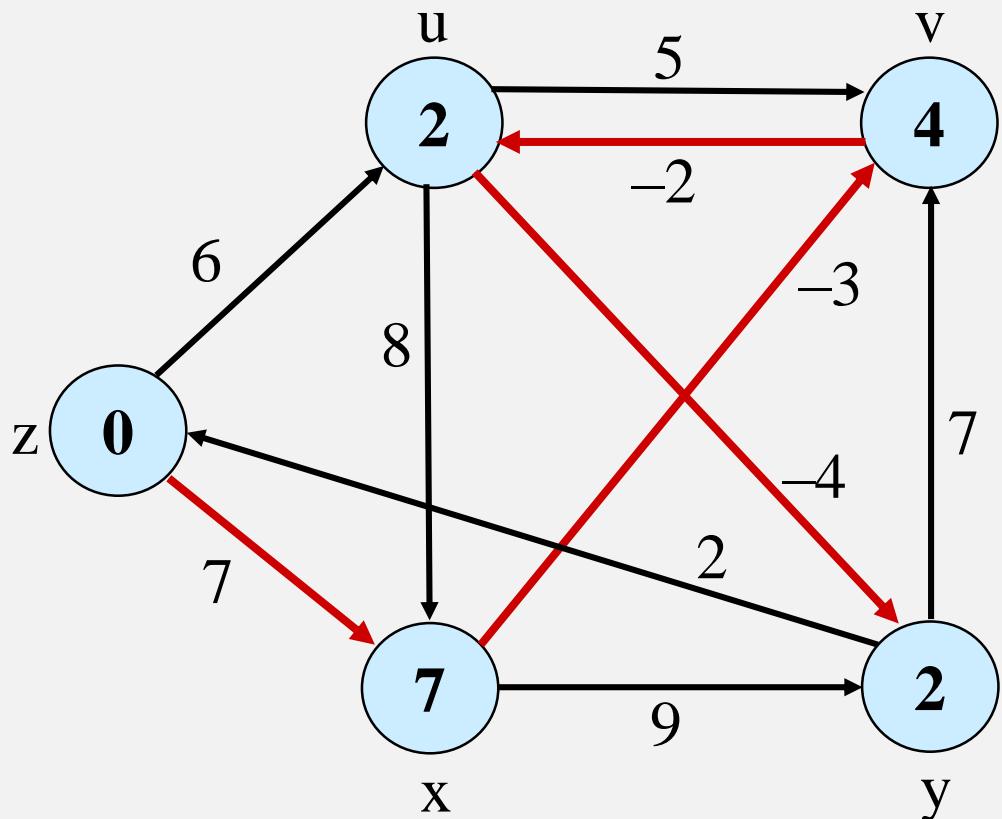
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end

```

Example



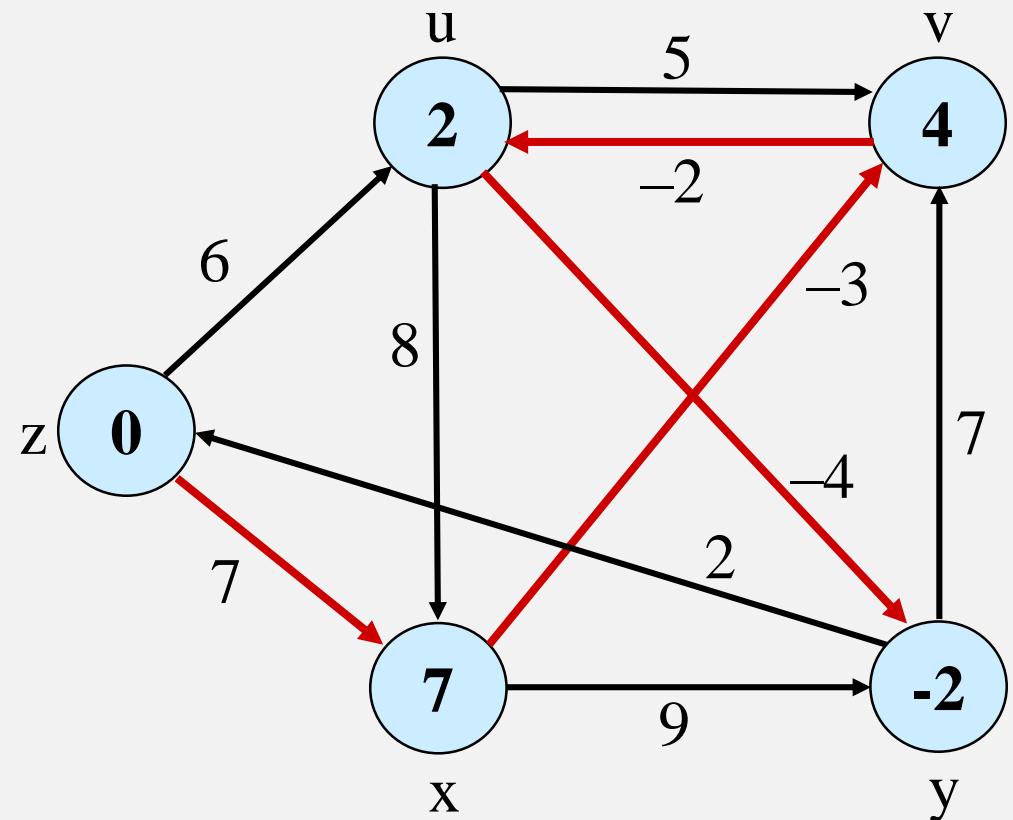
```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

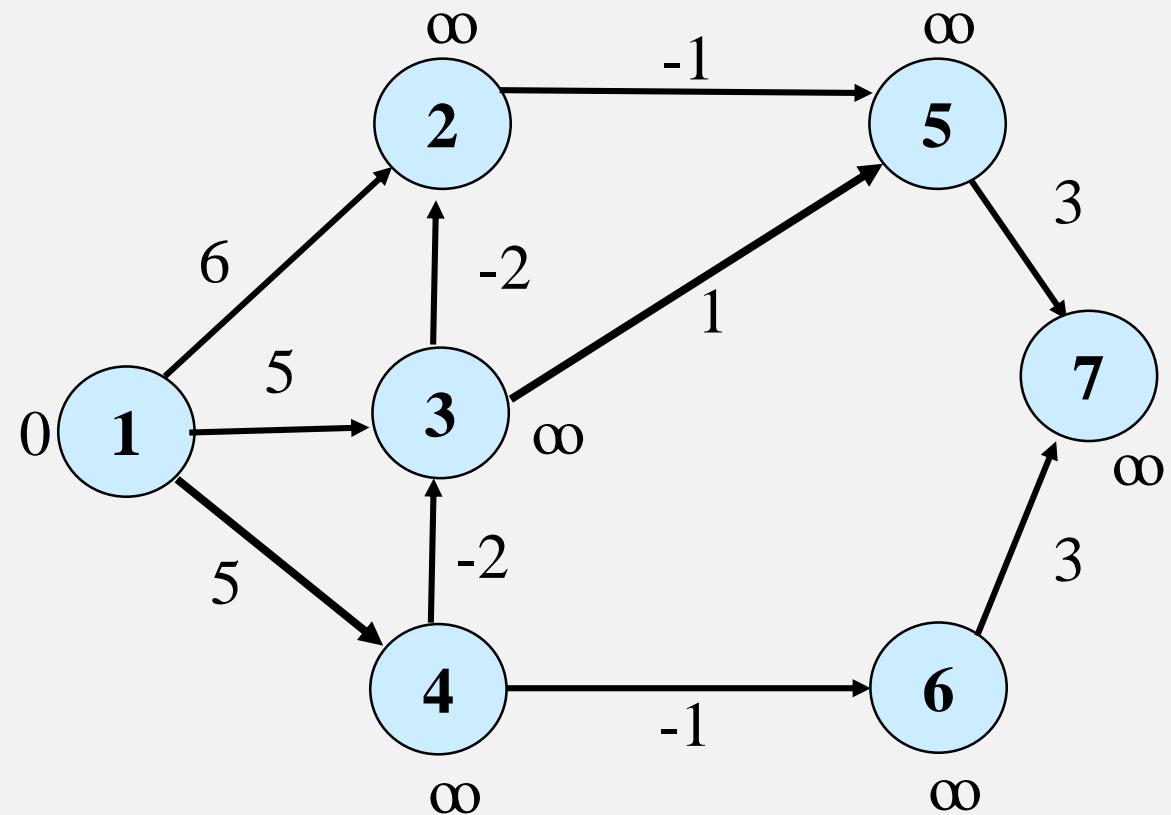
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end

```

Example



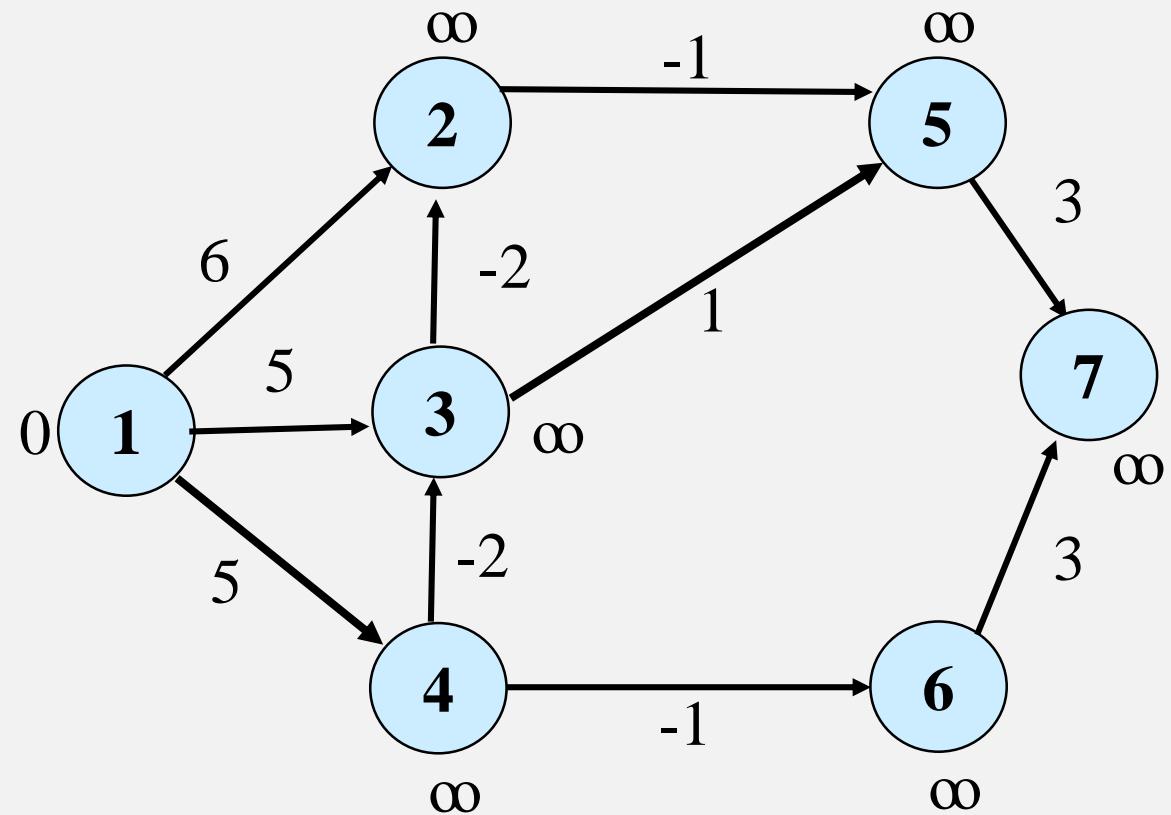
```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



```

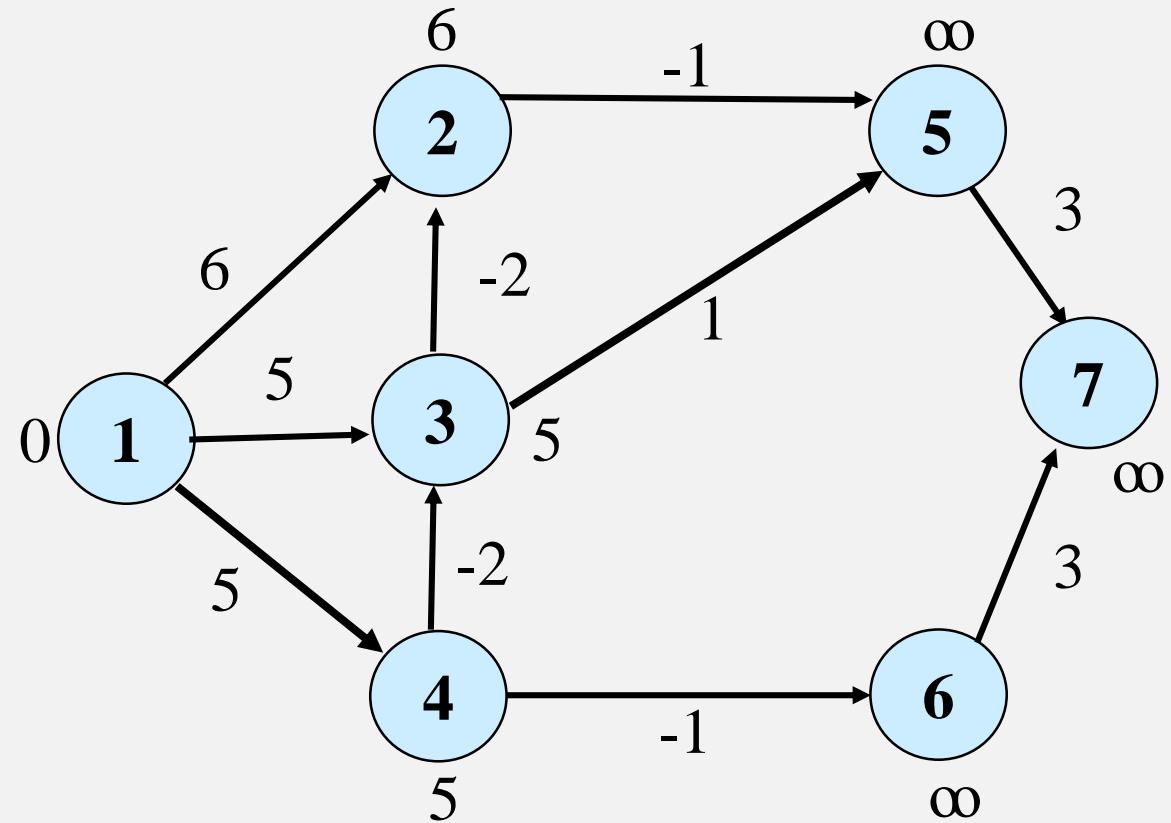
Initialize(G, s);
for i := 1 to |V[G]| - 1 do
  for each (u, v) in E[G] do
    Relax(u, v, w)
  end
end;
for each (u, v) in E[G] do
  if d[v] > d[u] + w(u, v) then
    return false
  end
end;
return true
  
```

```

Relax(u, v, w)
  if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
    π[v] := u
  end
  
```

$$E(G)=\{(1,2), (1,3), (1,4), (2,5), (3,5), (3,2), (4,3), (4,6), (5,7), (6,7)\}$$

Example



```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

```

```

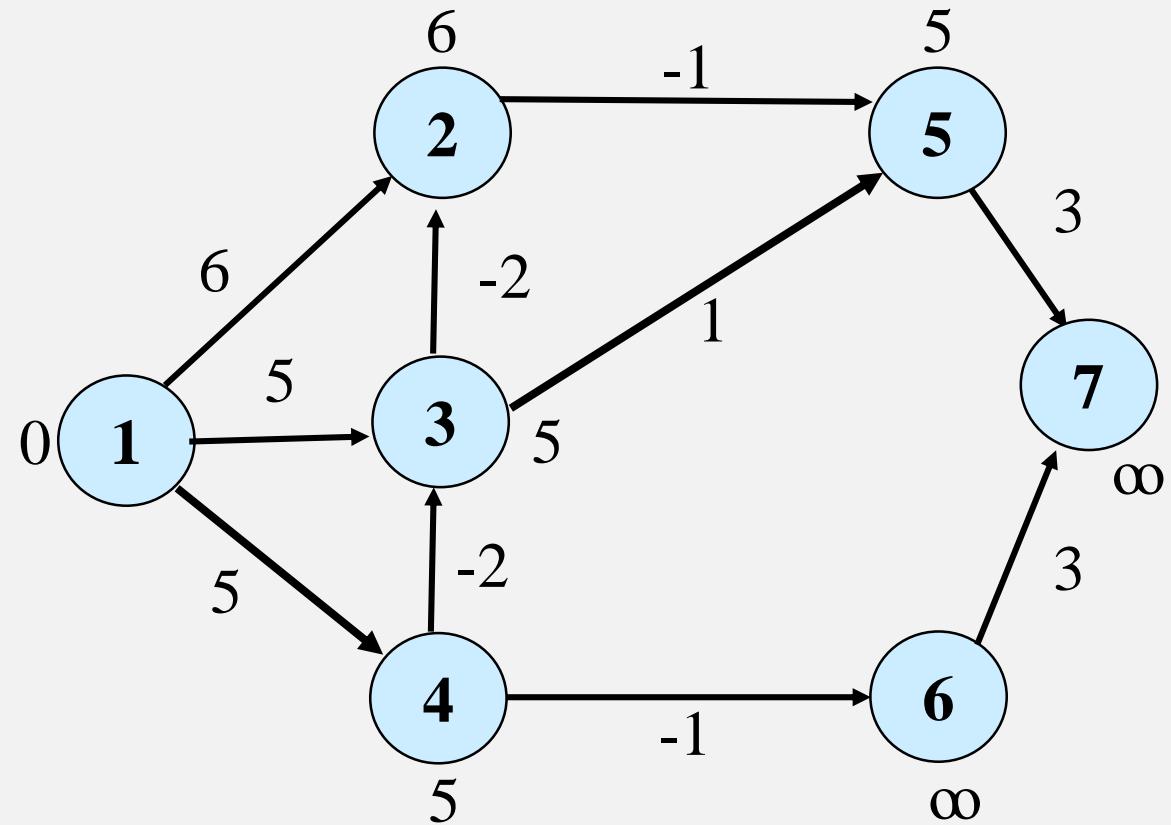
Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end

```

Iteration-1

$$E(G)=\{(1,2), (1,3), (1,4), (2,5), (3,5), (4,3), (4,7), (5,7), (6,7)\}$$

Example



```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

```

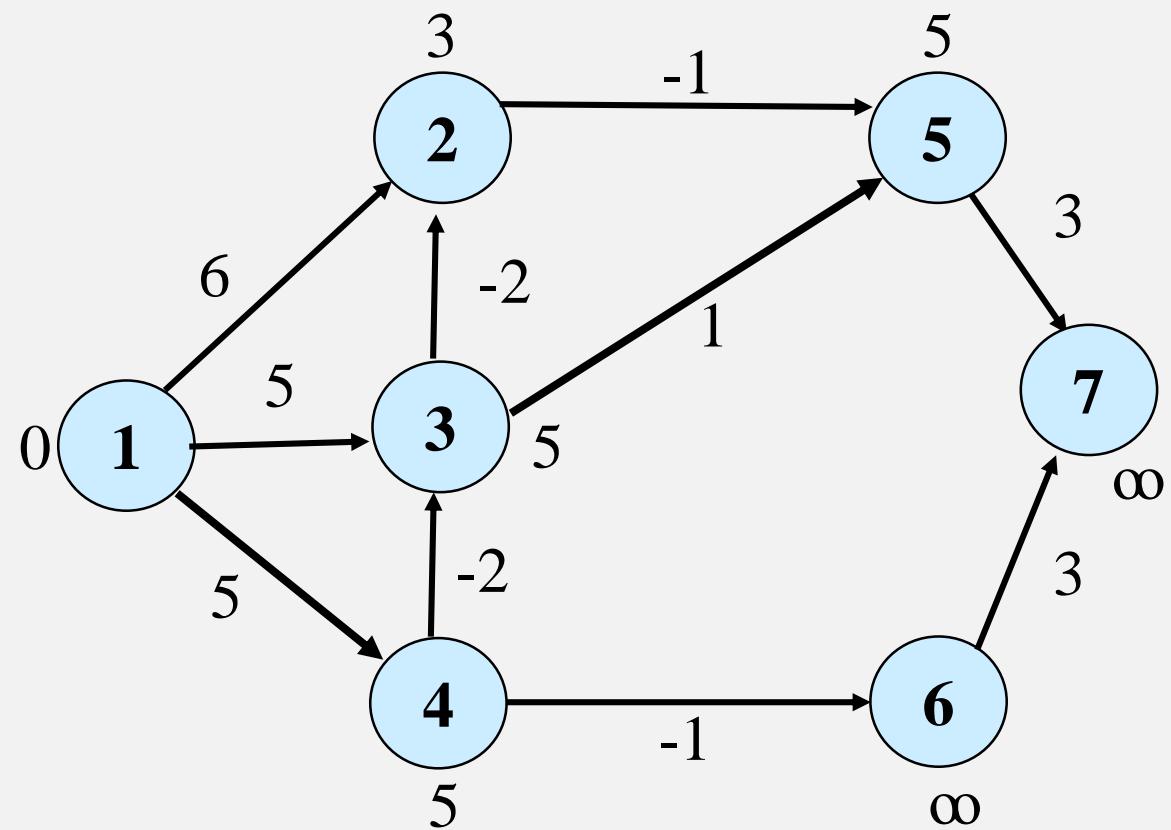
```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
     $\pi[v]$  := u
end

```

$$E(G) = \{(1,2), (1,3), (1,4), (2,5), (3,5), (3,2), (4,3), (4,6), (5,7), (6,7)\}$$

Example



```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

```

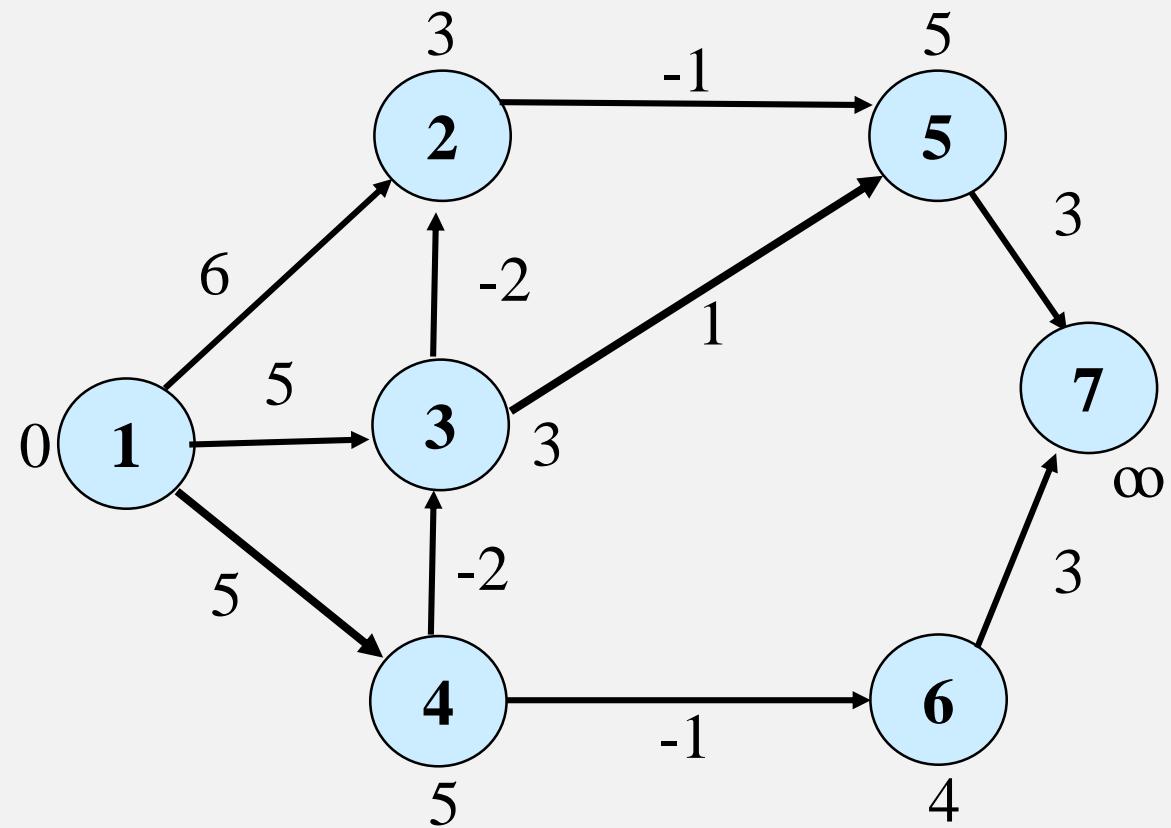
```

Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
     $\pi[v]$  := u
end

```

$$E(G)=\{(1,2), (1,3), (1,4), (2,5), (3,5), (3,2), (4,3), (4,6), (5,7), (6,7)\}$$

Example



```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

```

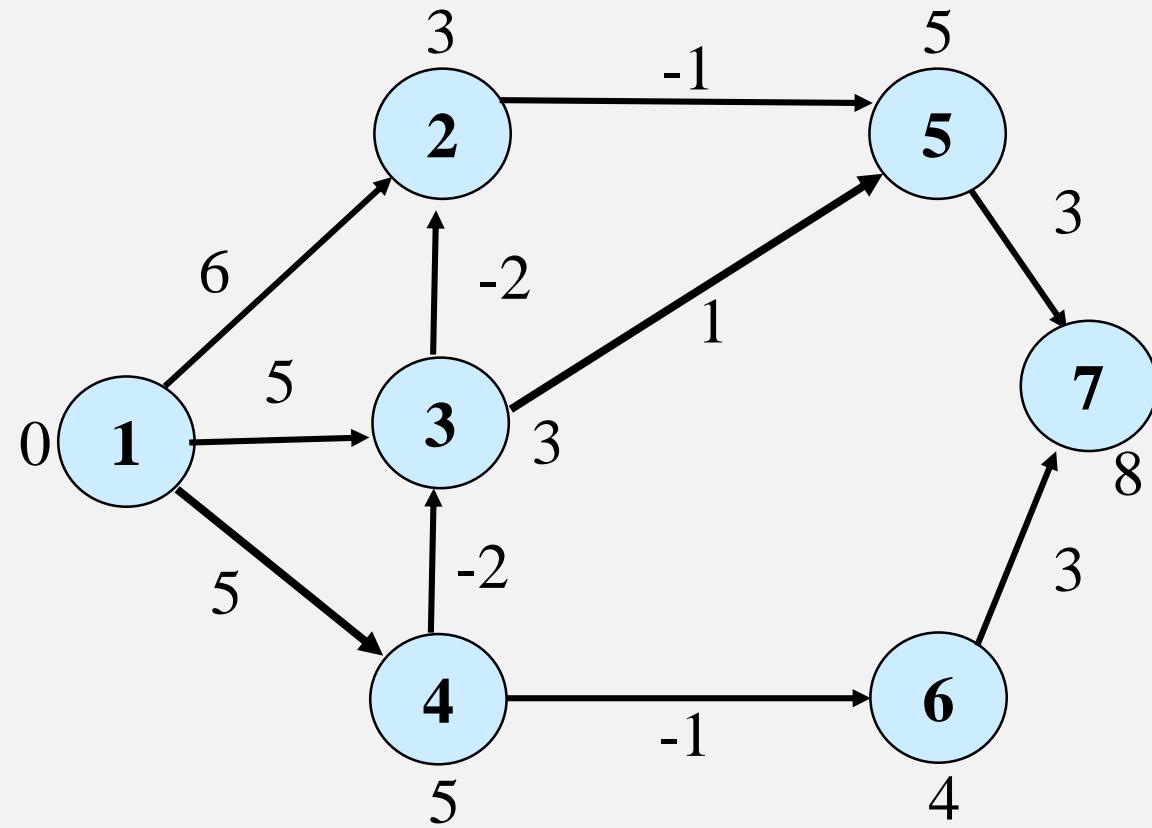
```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end

```

$$E(G)=\{(1,2), (1,3),(1,4), (2,5), (3,5),(3,2),(4,3),(4,6),(5,7),(6,7)\}$$

Example



$$E(G)=\{(1,2), (1,3),(1,4), (2,5), (3,5),(3,2),(4,3),(4,6), (5,7), (6,7)\}$$

```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

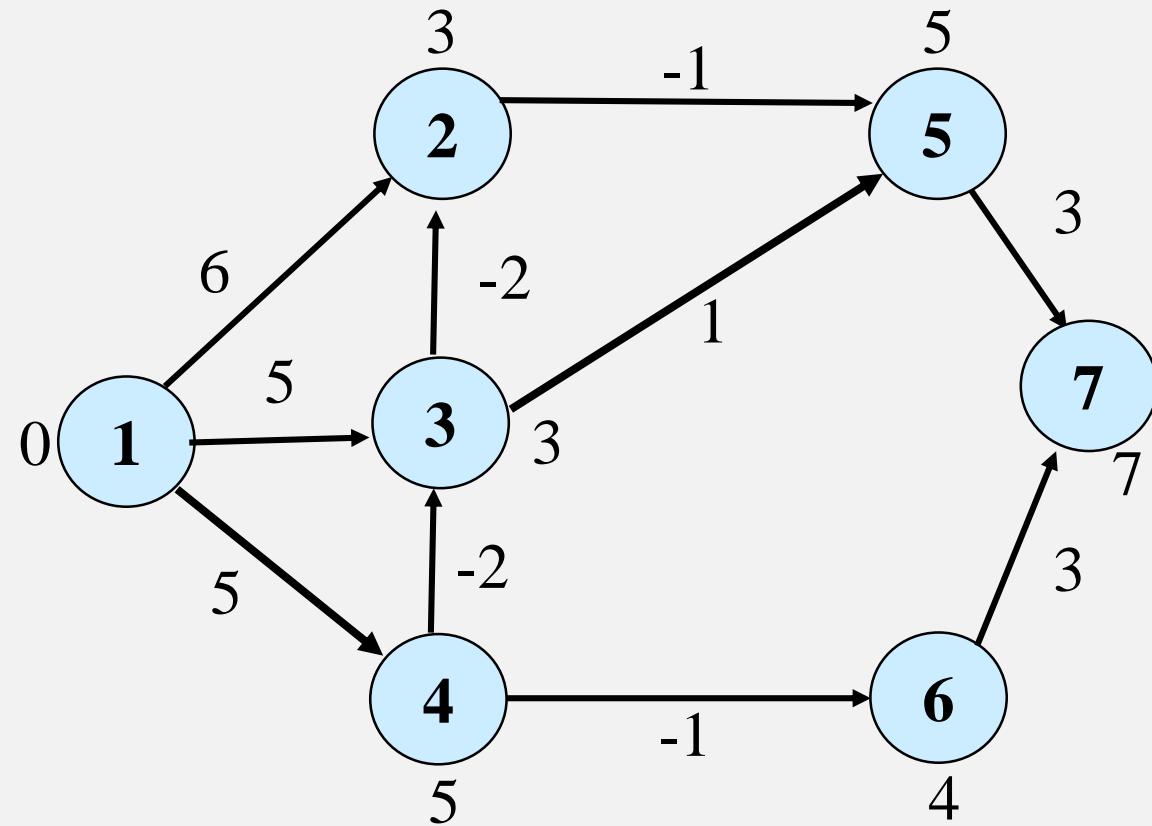
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end

```

Example



```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

```

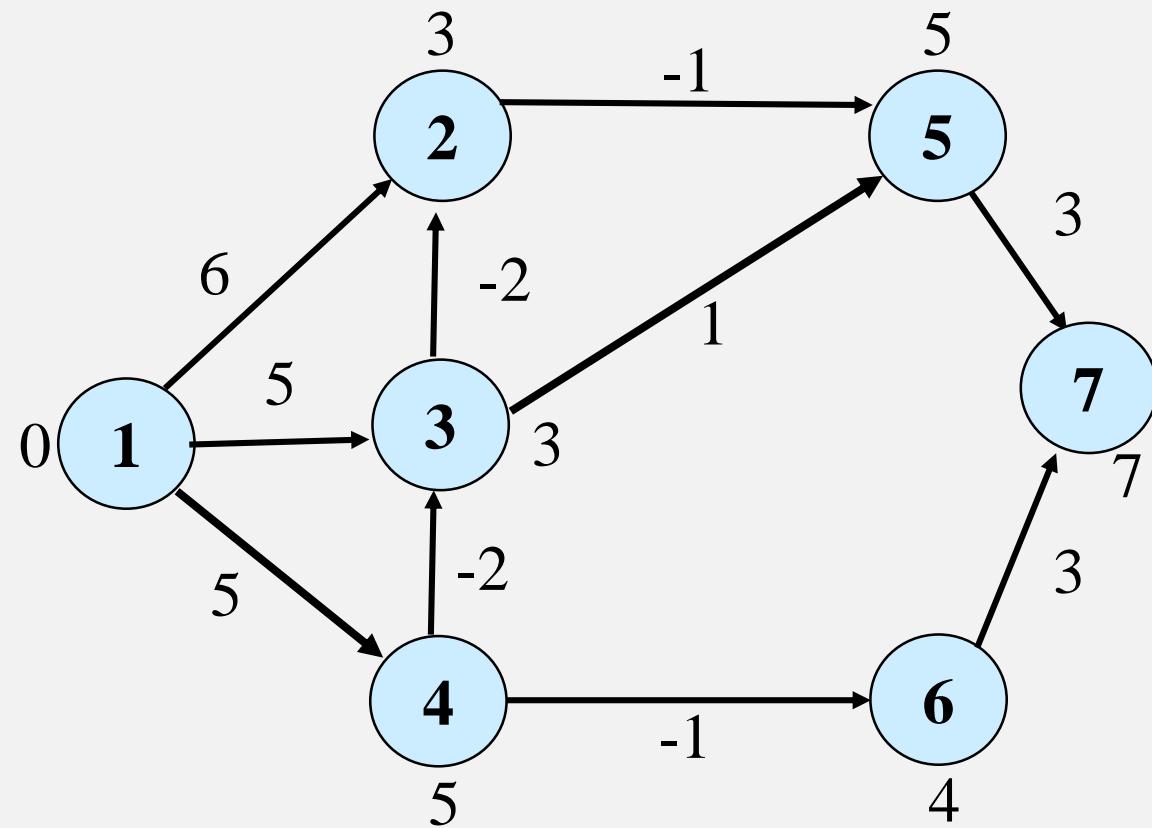
```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end

```

$$E(G)=\{(1,2), (1,3), (1,4), (2,5), (3,5), (3,2), (4,3), (4,6), (5,7), (6,7)\}$$

Example



Iteration-2

$$E(G)=\{(1,2), (1,3),(1,4), (2,5), (3,5),(3,2), (4,3),(4,6), (5,7), (6,7)\}$$

```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

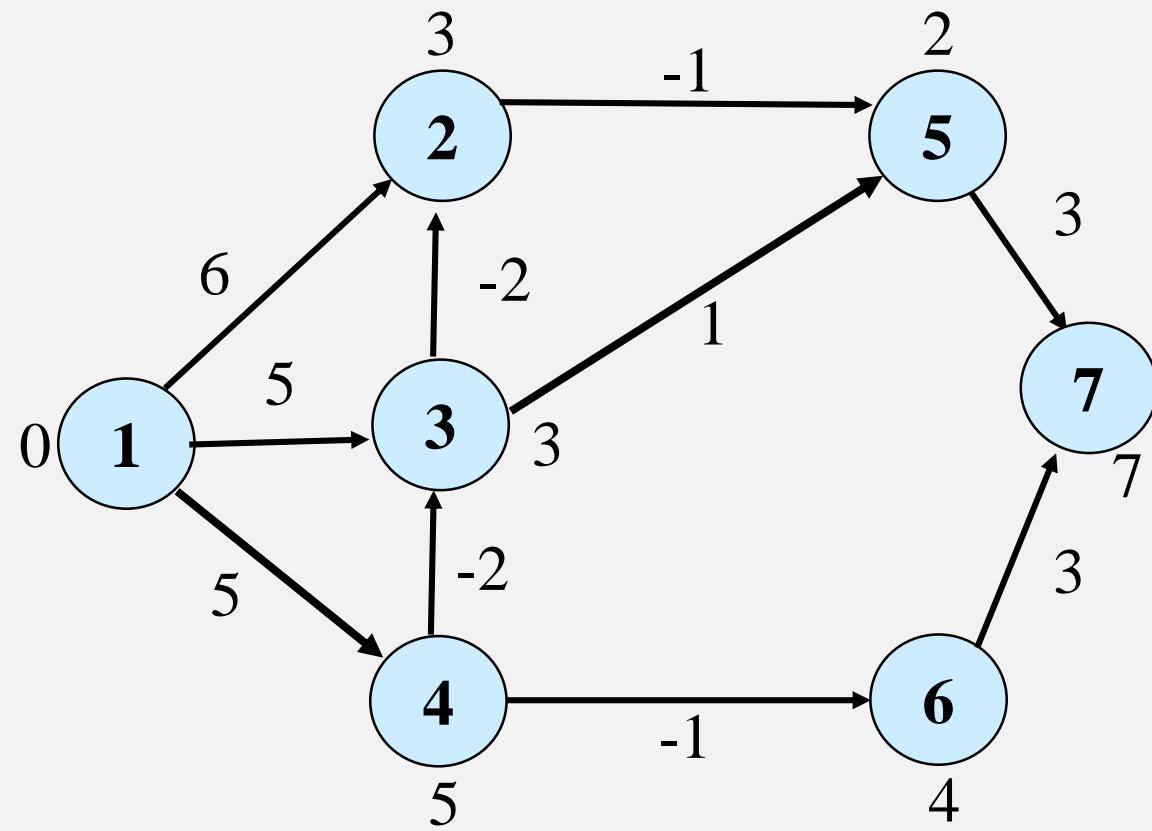
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end

```

Example



Iteration-2

$$E(G)=\{(1,2), (1,3),(1,4), (2,5), (3,5),(3,2), (4,3),(4,6), (5,7), (6,7)\}$$

```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

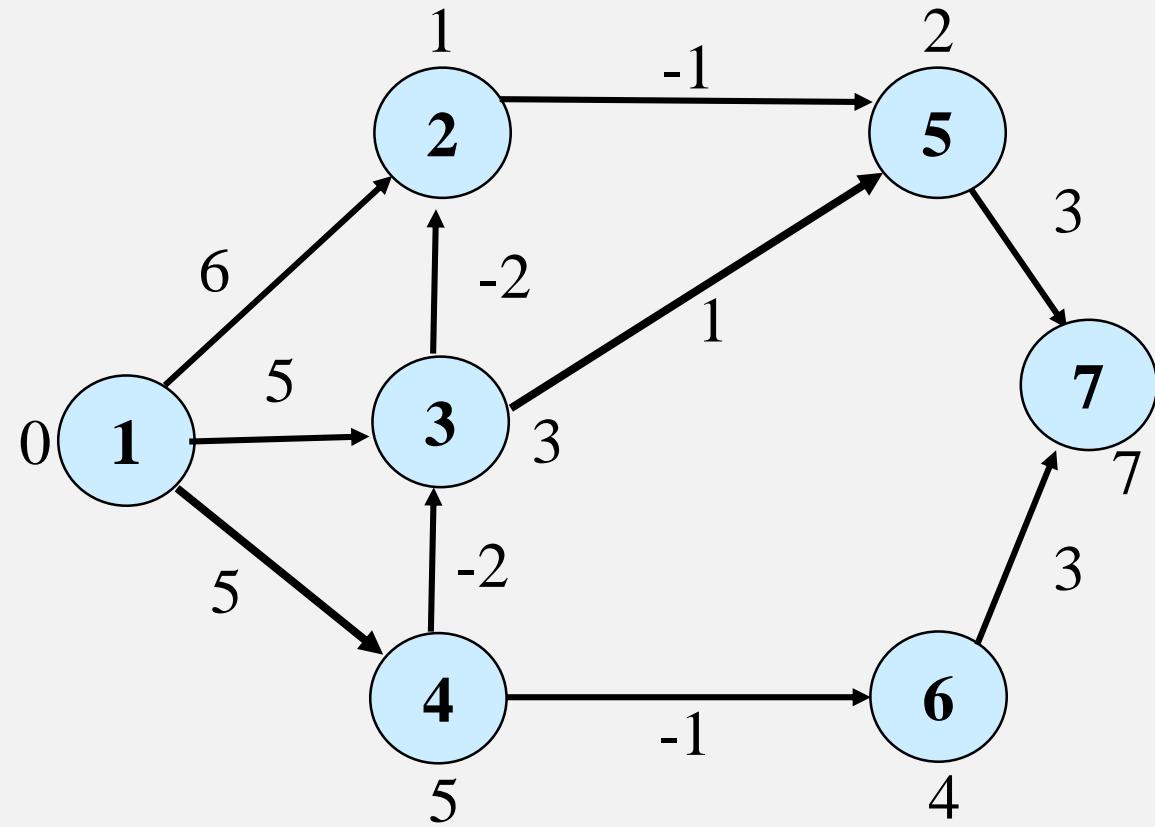
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end

```

Example



Iteration-2

$$E(G)=\{(1,2), (1,3),(1,4), (2,5), (3,5),(3,2), (4,3),(4,6), (5,7), (6,7)\}$$

```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

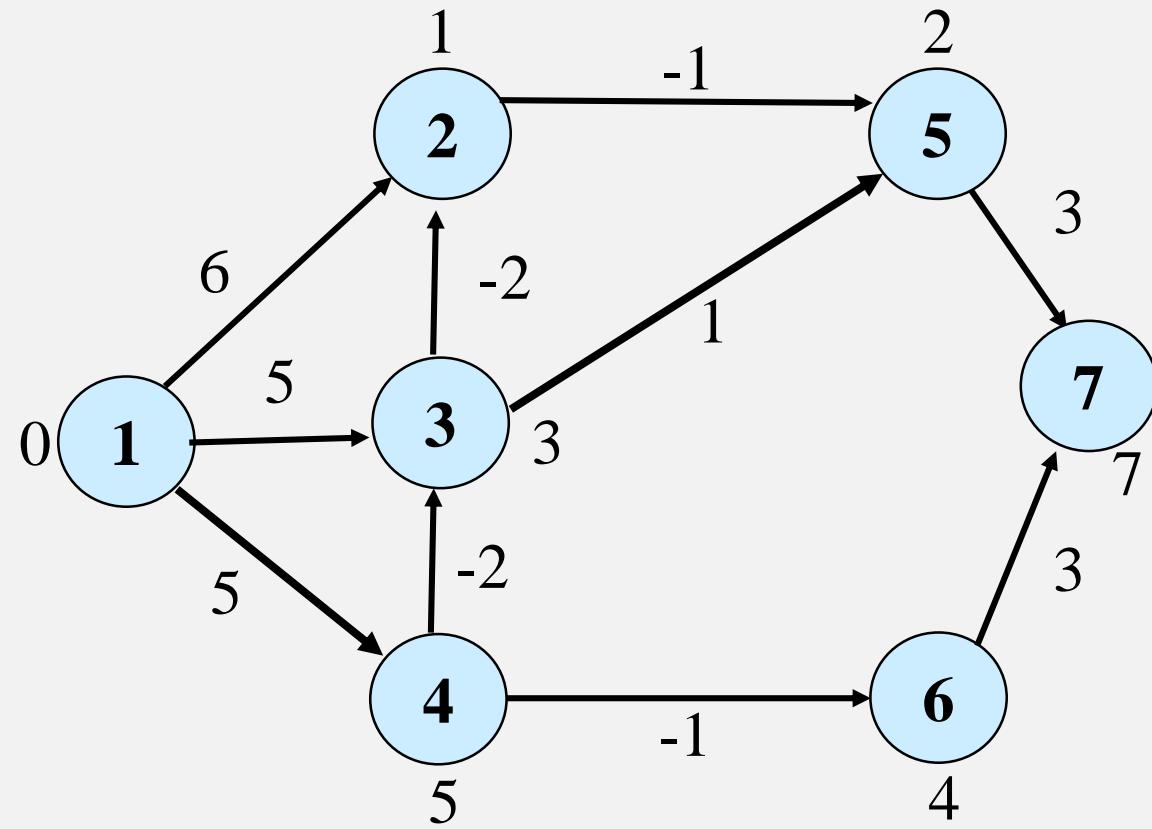
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end

```

Example



Iteration-2

$$E(G)=\{(1,2), (1,3), (1,4), (2,5), (3,5), (3,2), (4,3), (4,6), (5,7), (6,7)\}$$

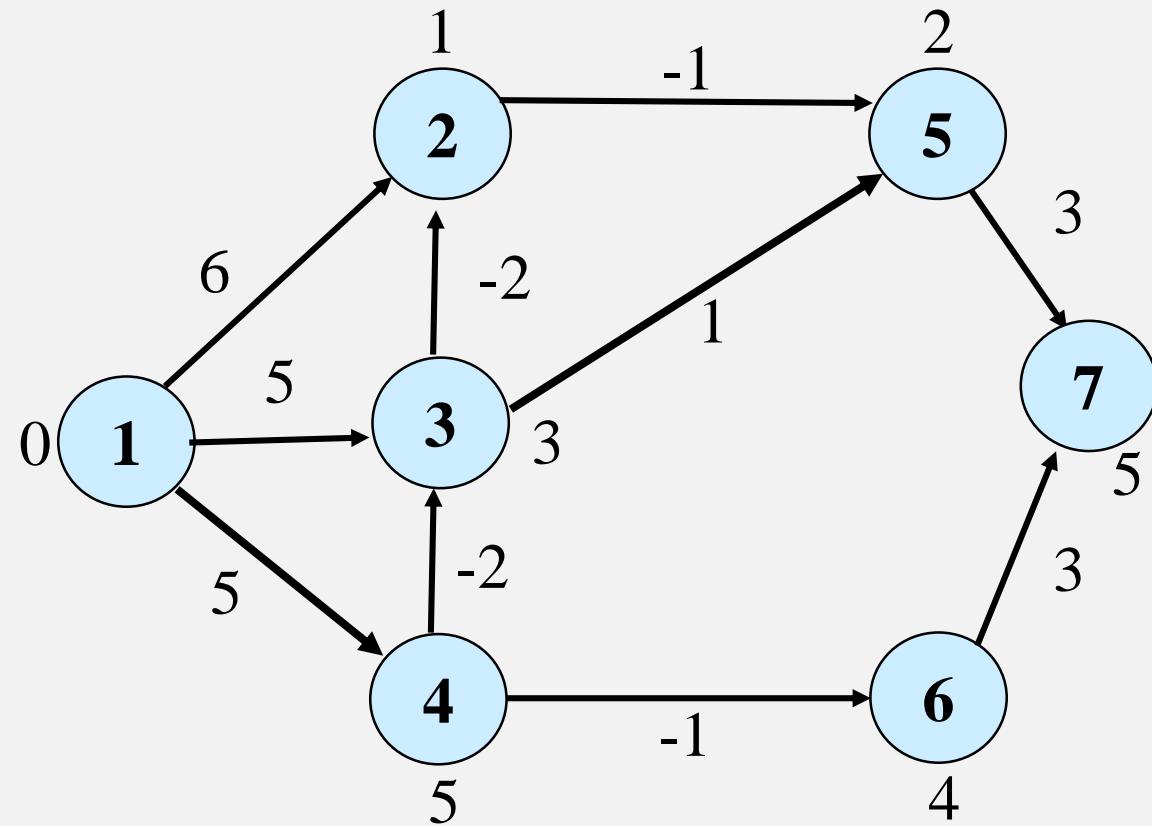
```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-2

$$E(G)=\{(1,2), (1,3), (1,4), (2,5), (3,5), (3,2), (4,3), (4,6), (5,7), (6,7)\}$$

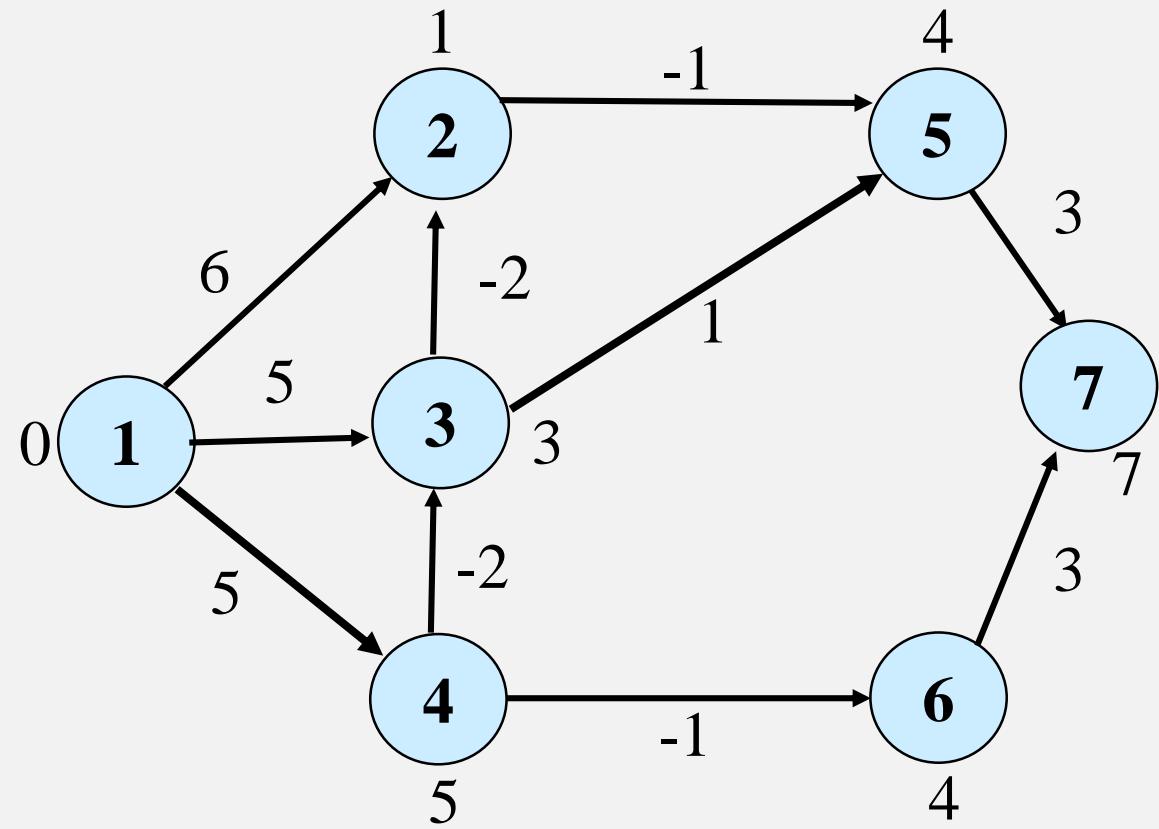
```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-2

$$E(G)=\{(1,2), (1,3),(1,4), (2,5), (3,5),(3,2), (4,3),(4,6), (5,7), (6,7)\}$$

```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

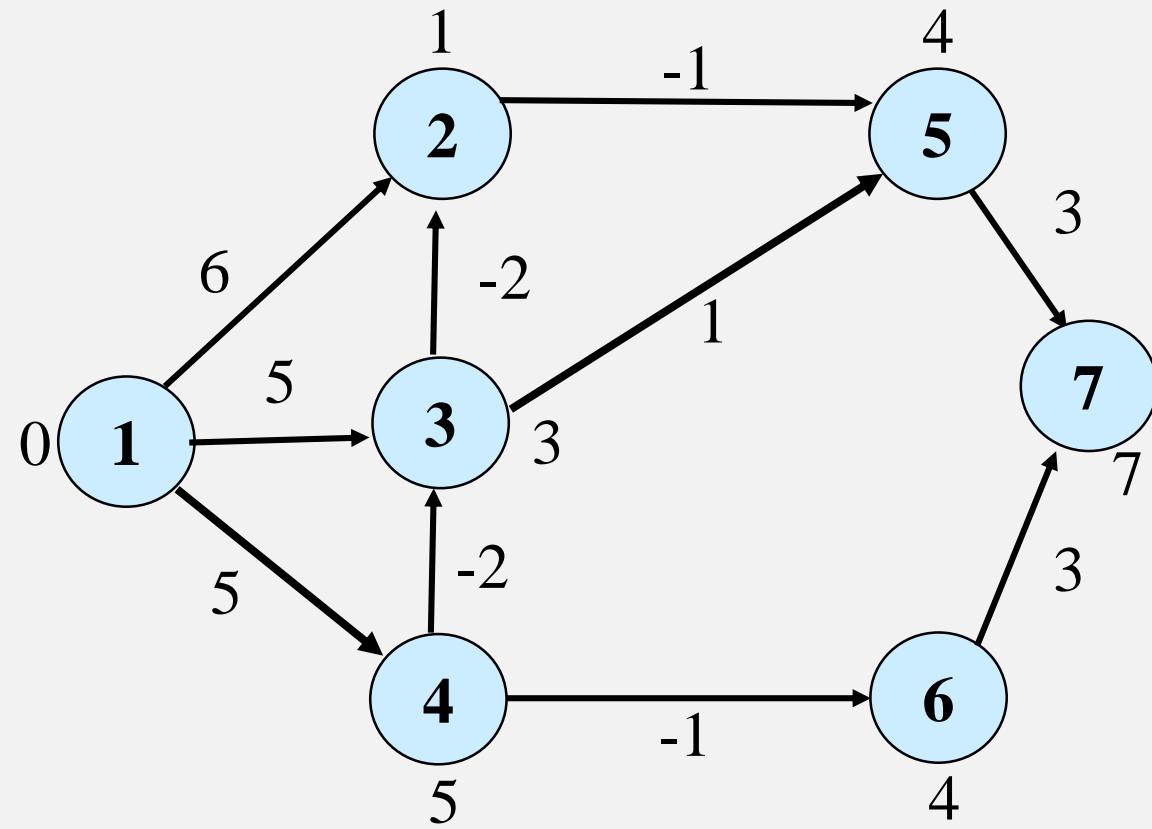
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end

```

Example



Iteration-3

$$E(G)=\{(1,2), (1,3),(1,4), (2,5), (3,5),(3,2), (4,3),(4,6), (5,7), (6,7)\}$$

```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

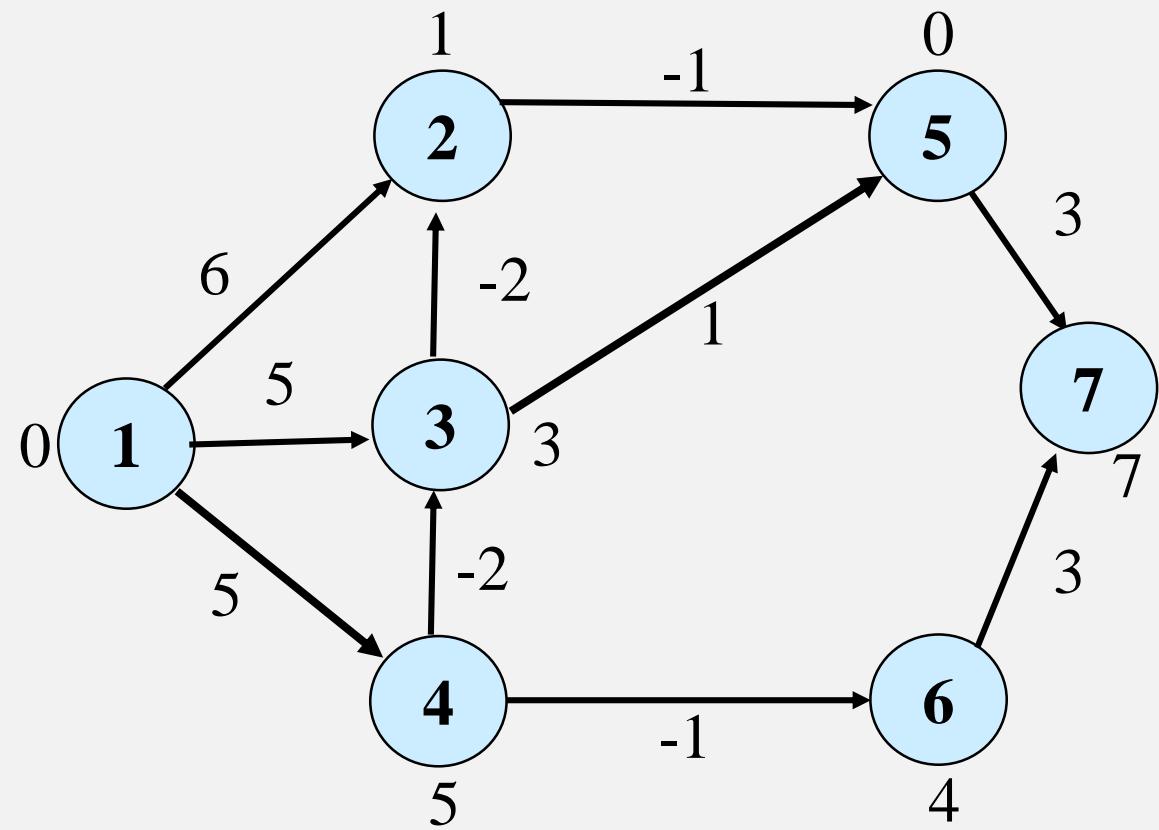
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end

```

Example



Iteration-3

$$E(G)=\{(1,2), (1,3), (1,4), (2,5), (3,5), (3,2), (4,3), (4,6), (5,7), (6,7)\}$$

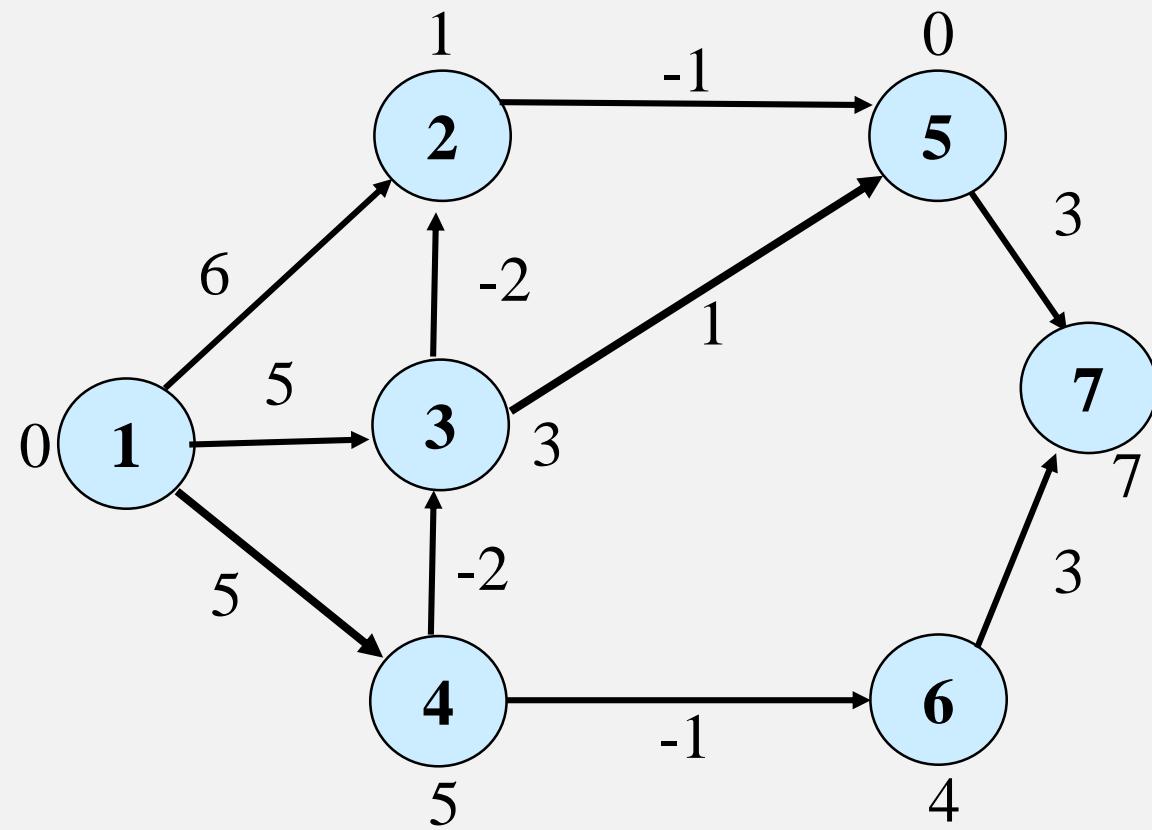
```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-3

$$E(G)=\{(1,2), (1,3), (1,4), (2,5), (3,5), (3,2), (4,3), (4,6), (5,7), (6,7)\}$$

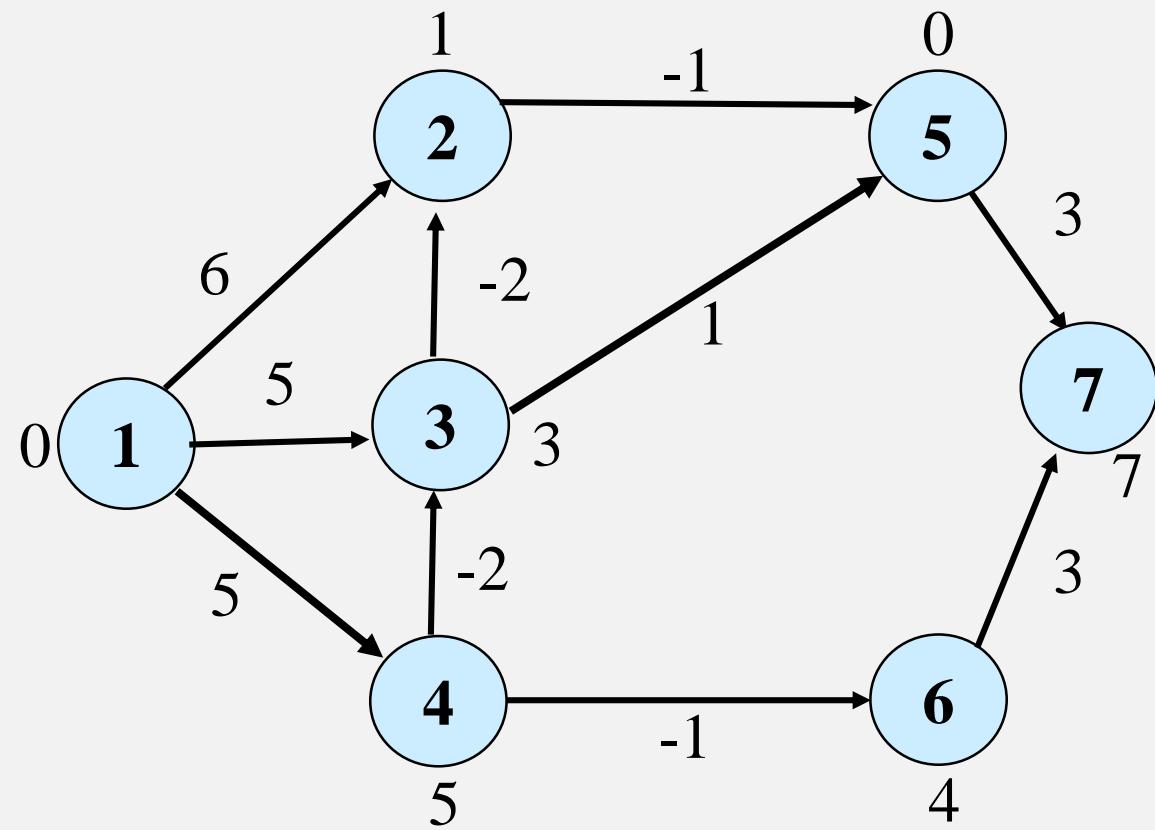
```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-3

$$E(G)=\{(1,2), (1,3), (1,4), (2,5), (3,5), (3,2), (4,3), (4,6), (5,7), (6,7)\}$$

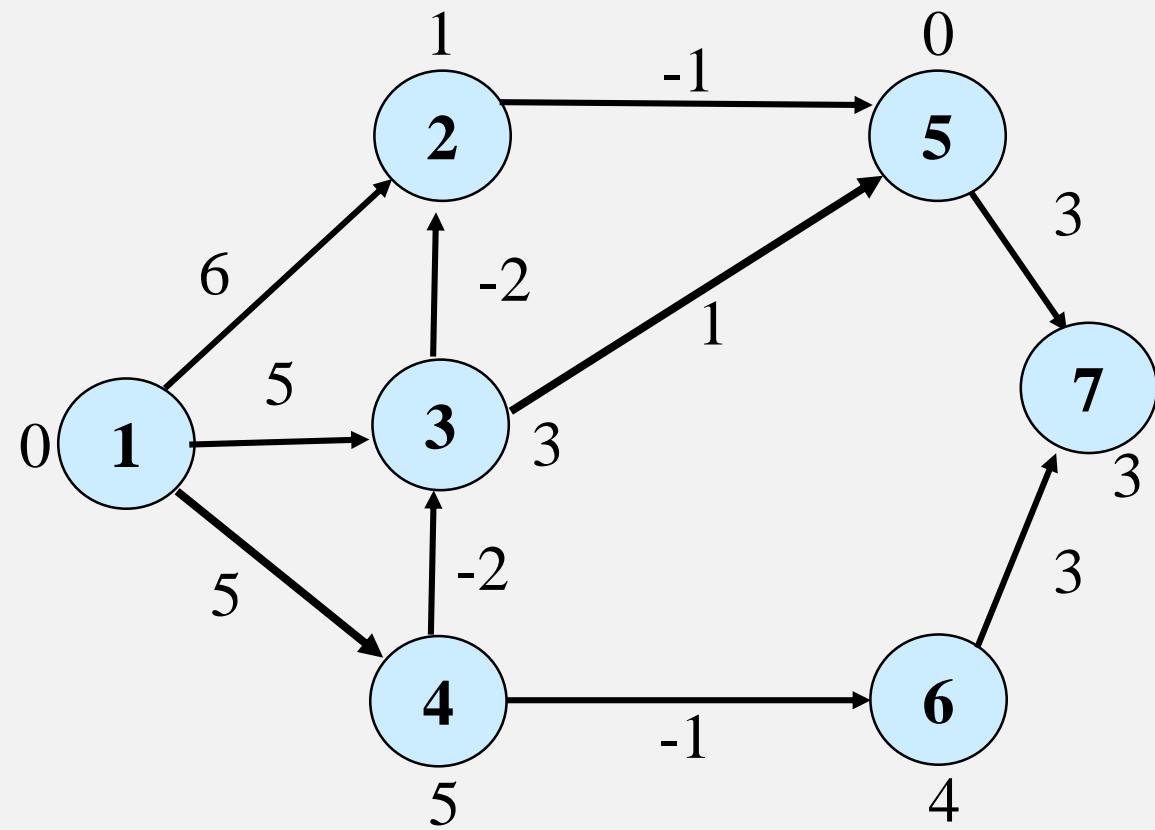
```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-3

$$E(G)=\{(1,2), (1,3),(1,4), (2,5), (3,5),(3,2), (4,3),(4,6), (5,7), (6,7)\}$$

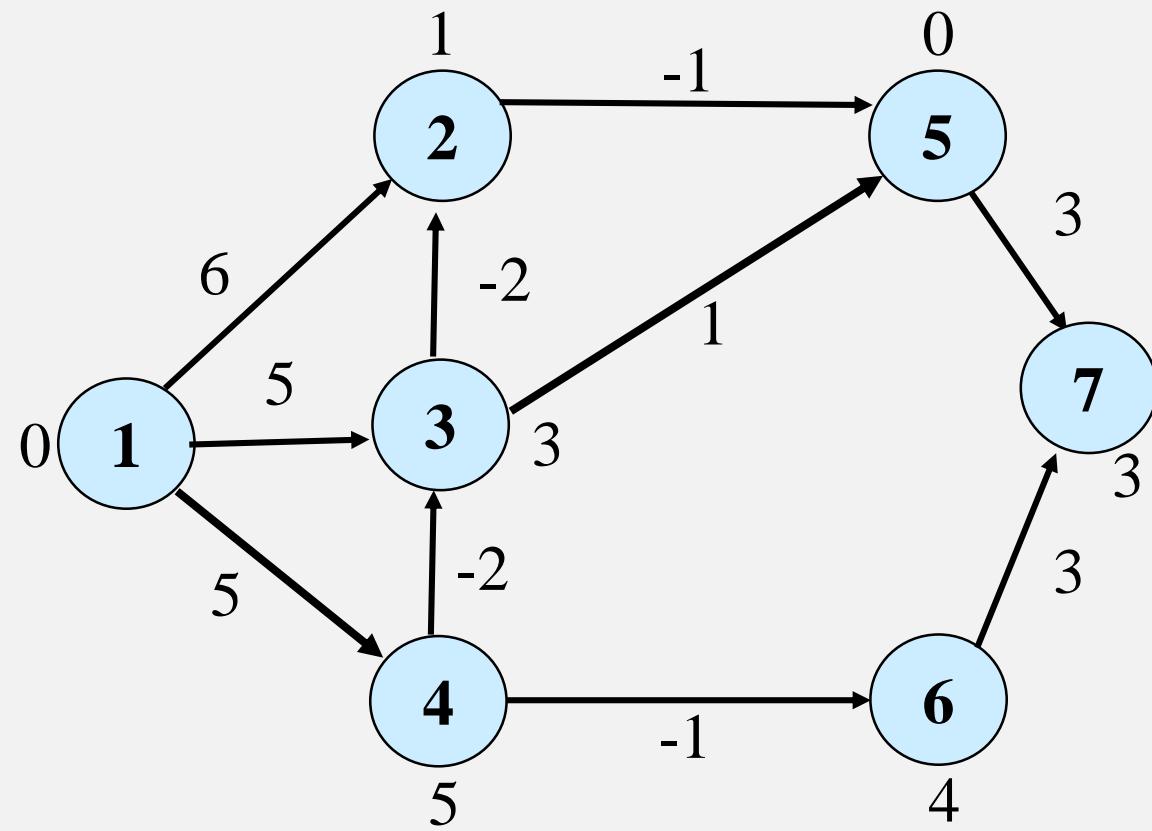
```

Initialize(G, s);
for i := 1 to |V[G]| – 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-3

$$E(G)=\{(1,2), (1,3), (1,4), (2,5), (3,5), (3,2), (4,3), (4,6), (5,7), (6,7)\}$$

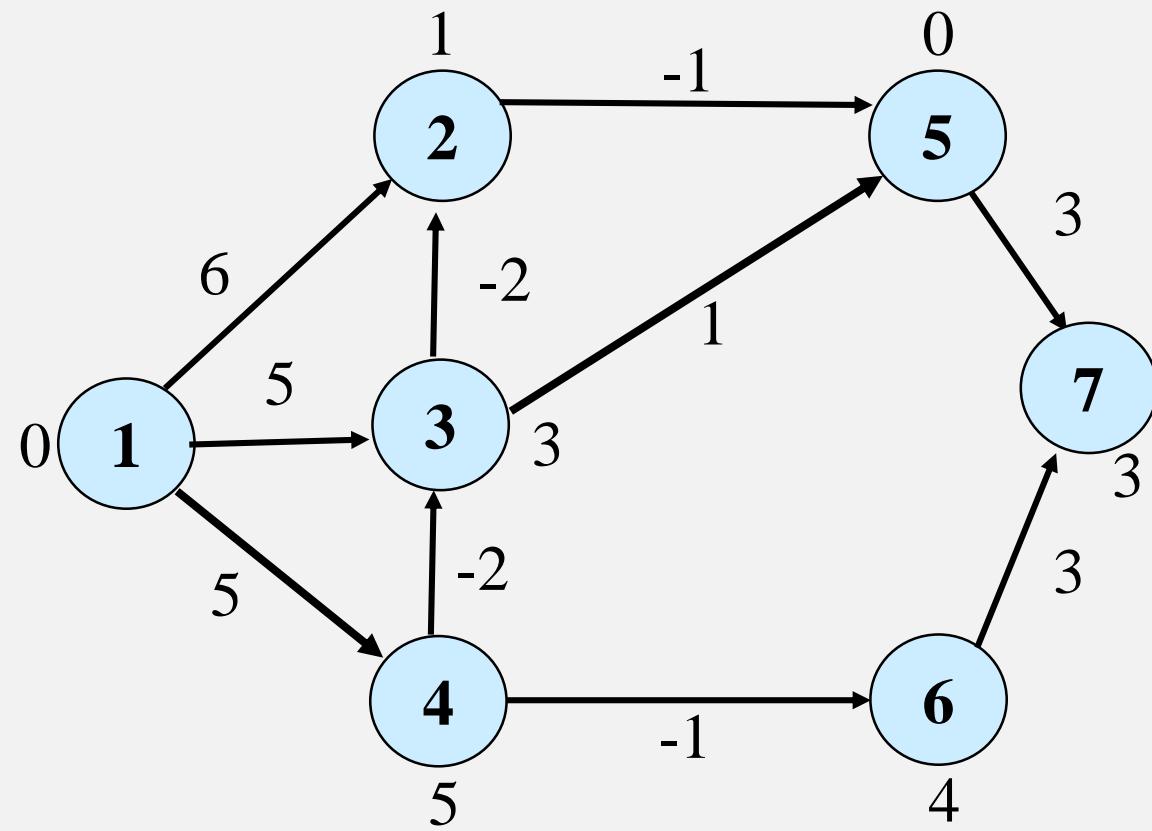
```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

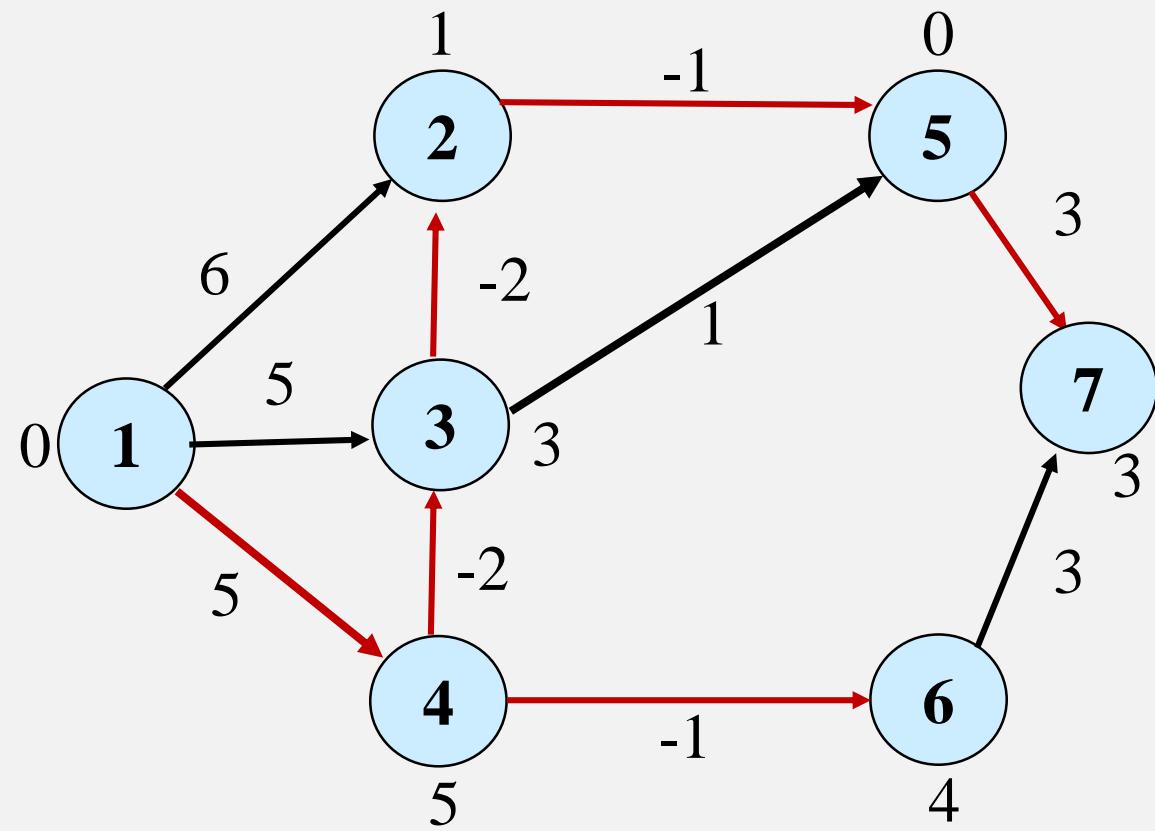
```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Iteration-4 Repeat the process for the given set of edges.

$$E(G)=\{(1,2), (1,3), (1,4), (2,5), (3,5), (3,2), (4,3), (4,6), (5,7), (6,7)\}$$

Example



Iteration-4

$$E(G)=\{(1,2), (1,3), (1,4), (2,5), (3,5), (3,2), (4,3), (4,6), (5,7), (6,7)\}$$

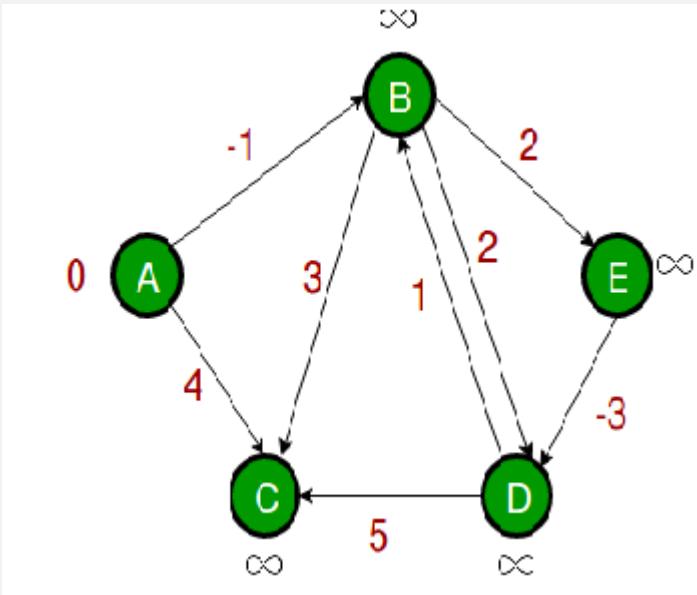
```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Set of Edges- ??
 Iteration- ??

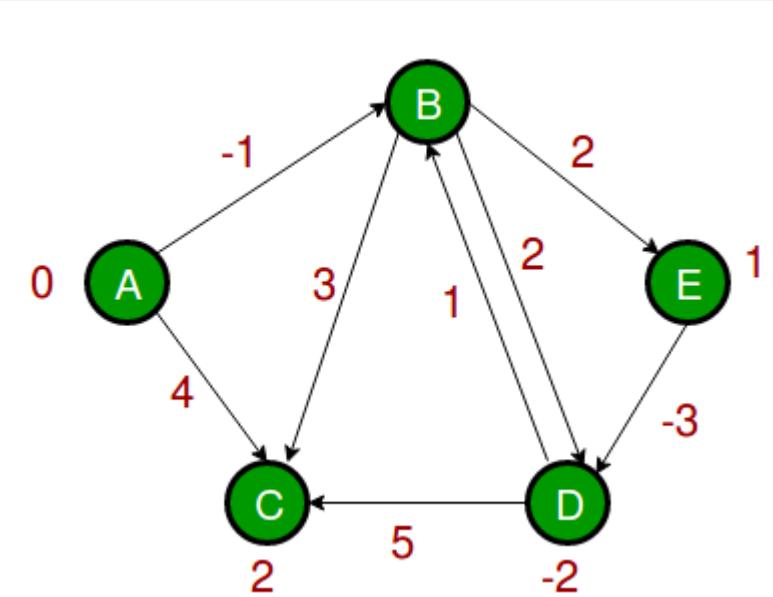
```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
  for each (u, v) in E[G] do
    Relax(u, v, w)
  end
end;
for each (u, v) in E[G] do
  if d[v] > d[u] + w(u, v) then
    return false
  end
end;
return true
  
```

```

Relax(u, v, w)
  if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
    π[v] := u
  end
  
```

Example



```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true

```

```

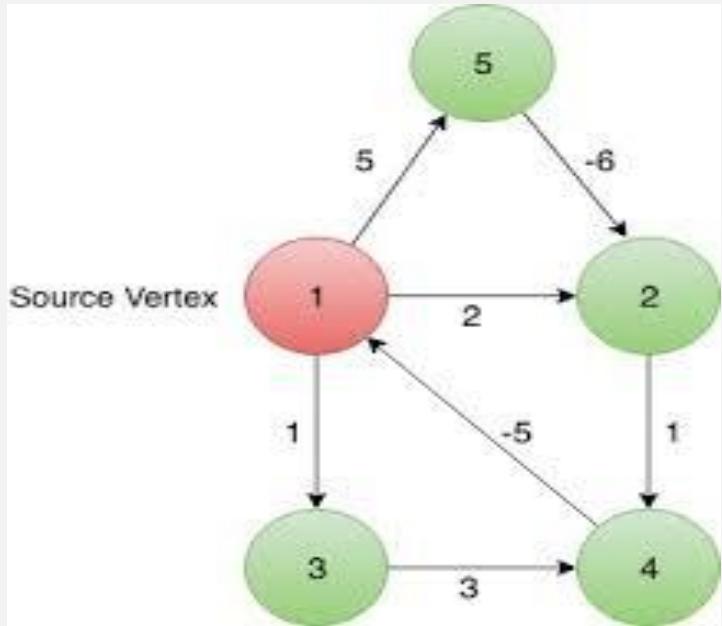
Relax(u, v, w)
if d[v] > d[u] + w(u, v) then
    d[v] := d[u] + w(u, v);
    π[v] := u
end

```

$$E(G)=\{(A,B), (A,C), (B,C), (B,D), (B,E), (D,B), (D,C), (E,D)\}$$

Iteration-??

Example



Set of edges-??

Iteration-??

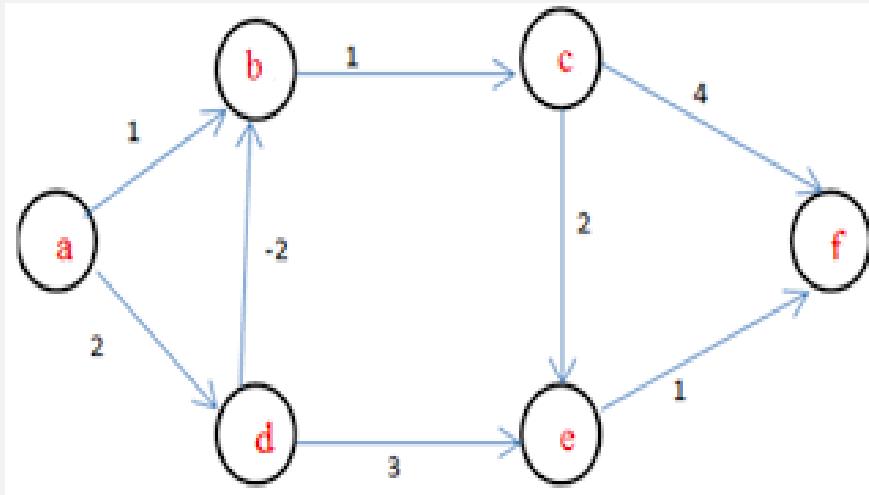
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Set of edges-??

Iteration-??

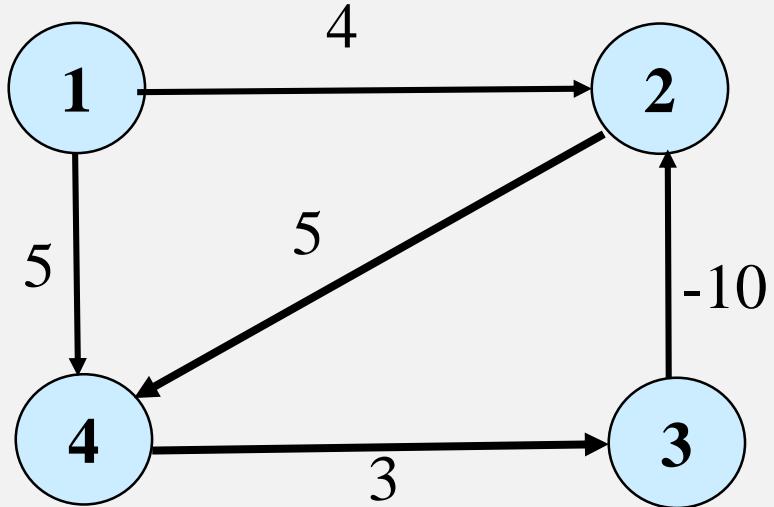
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-1

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

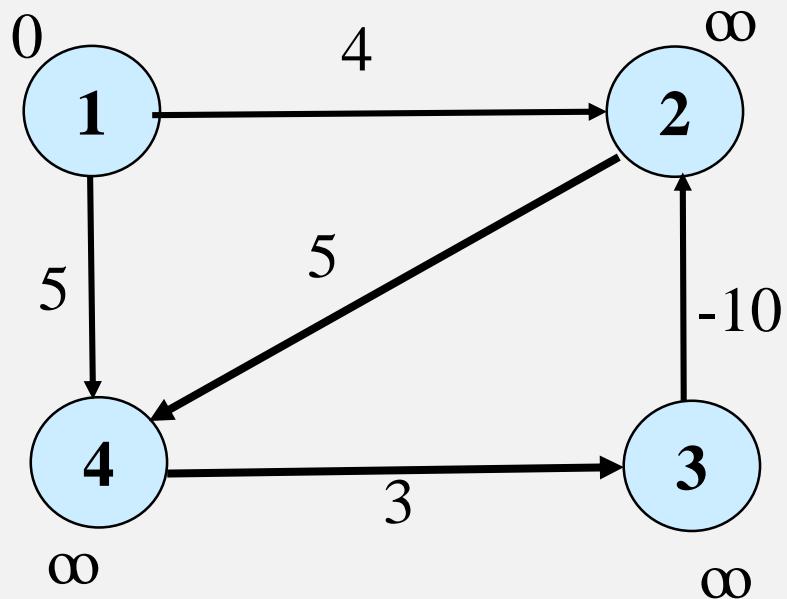
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-1

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

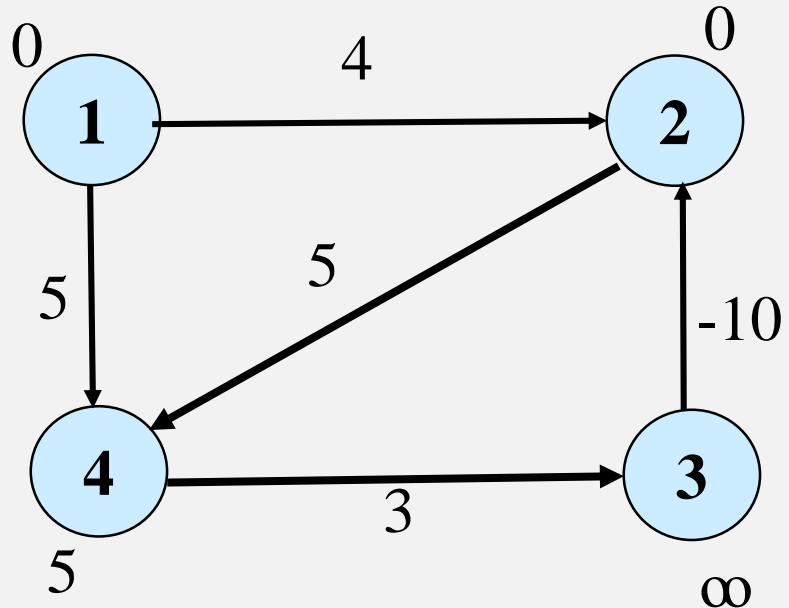
```

Initialize(G, s);
for i := 1 to |V[G]| - 1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-1

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,1)\}$$

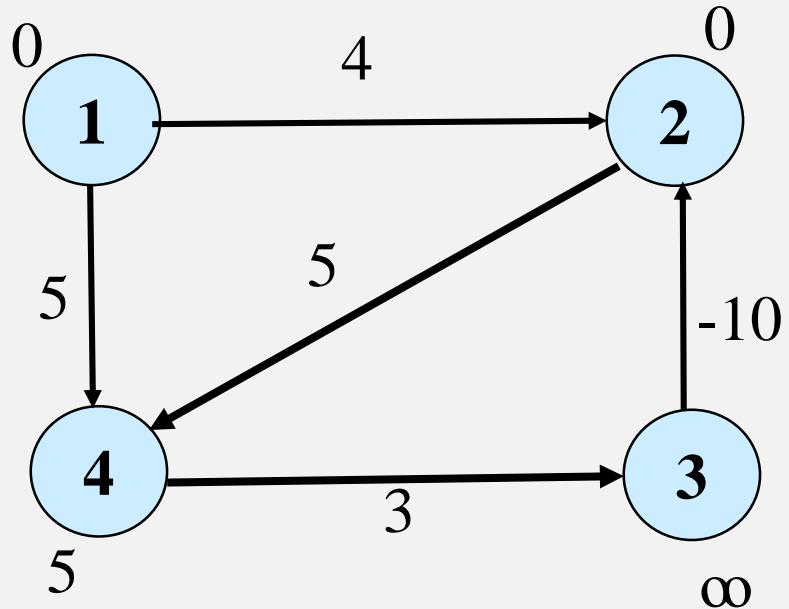
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-1

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

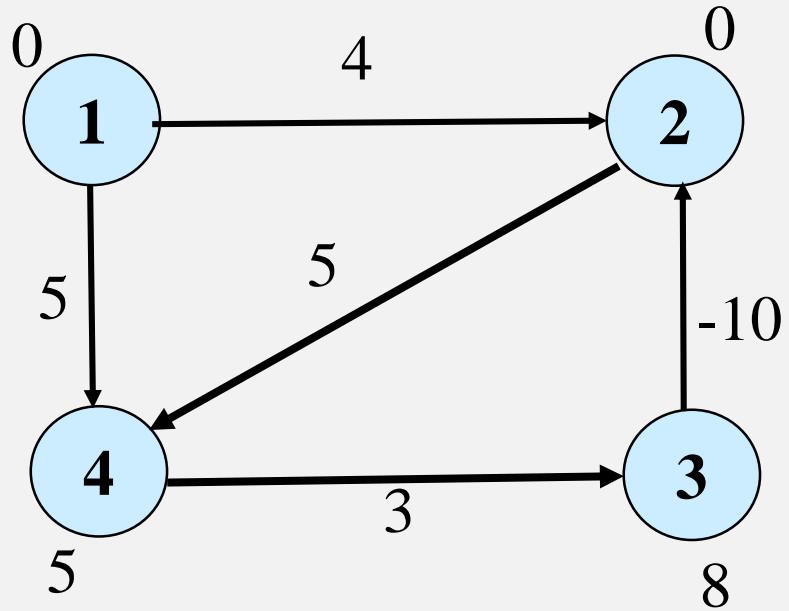
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-1

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

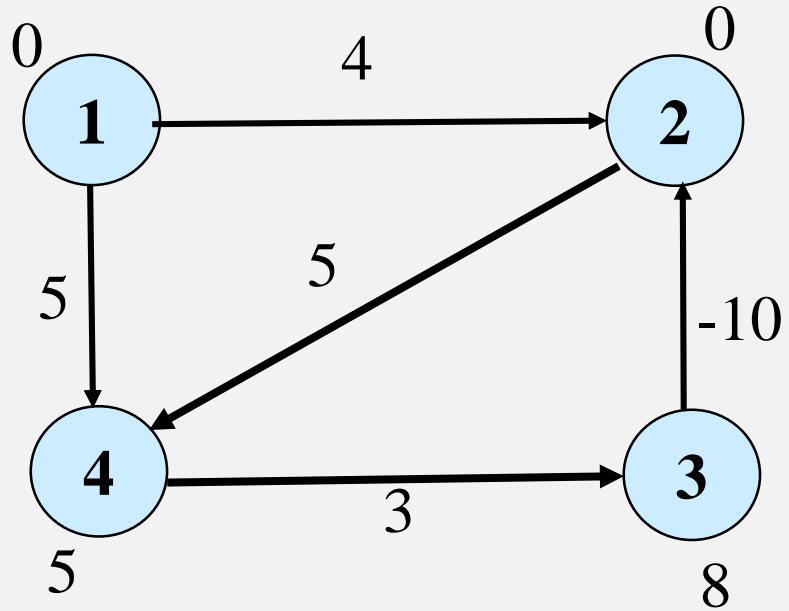
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-2

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

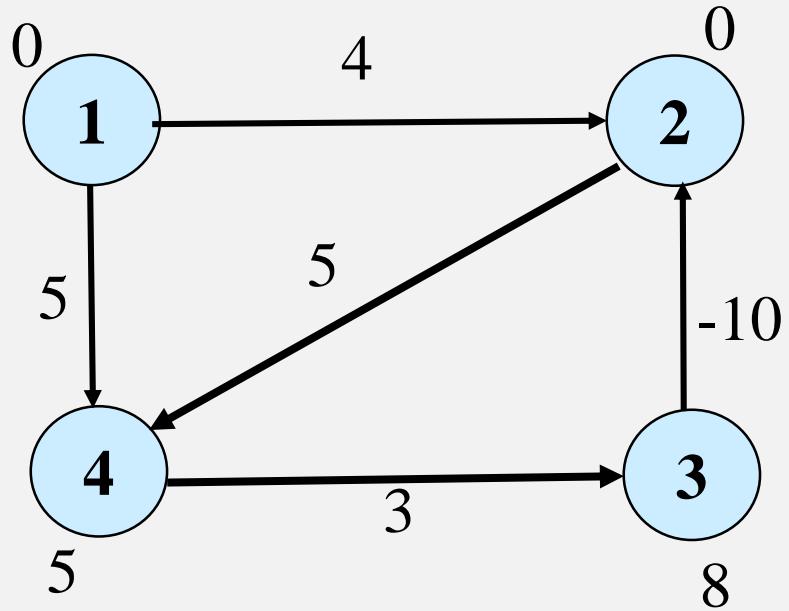
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
    
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
    
```

Example



Iteration-2

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

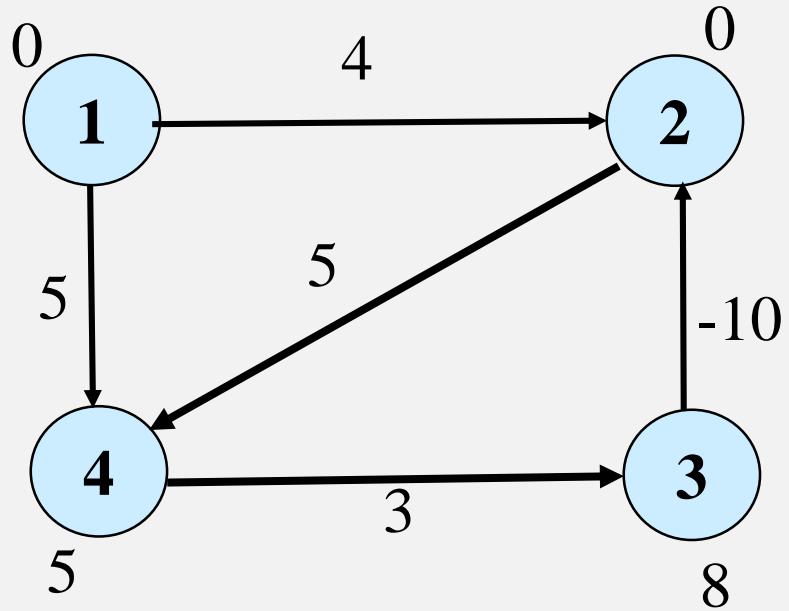
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-2

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

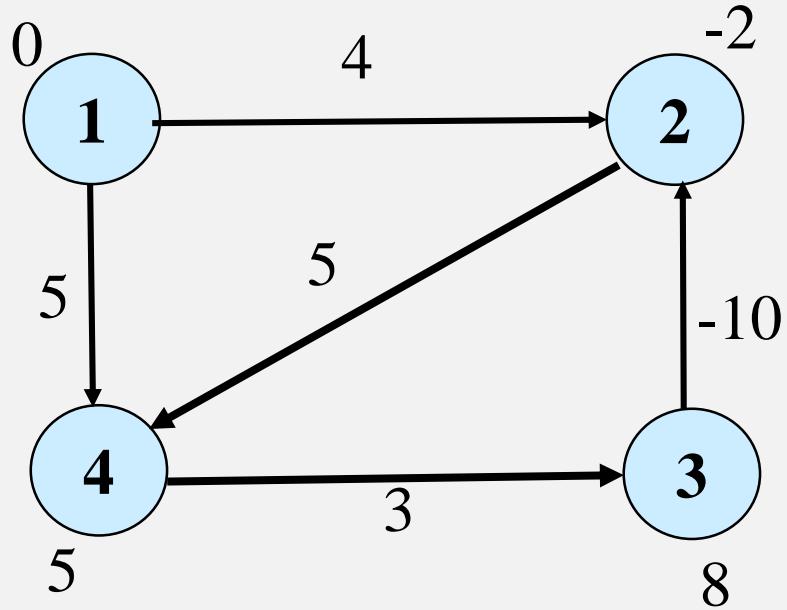
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
    
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
    
```

Example



Iteration-2

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

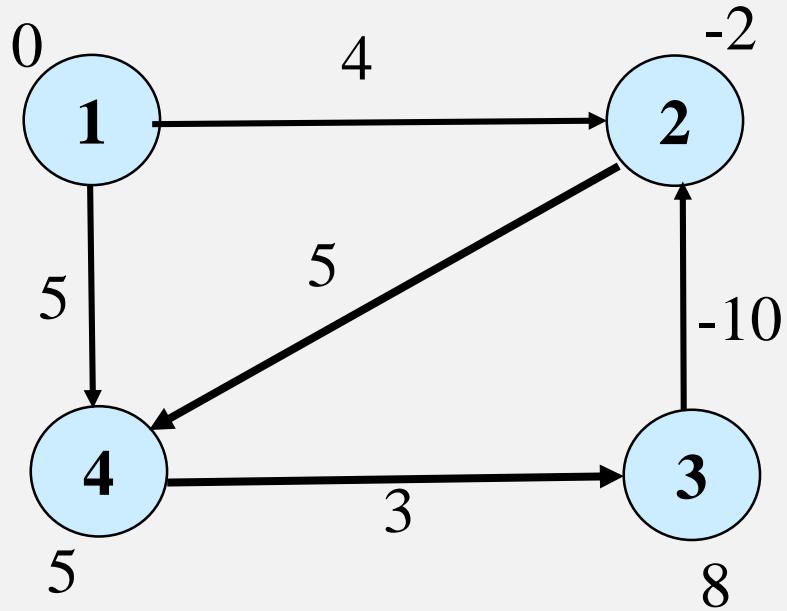
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-2

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

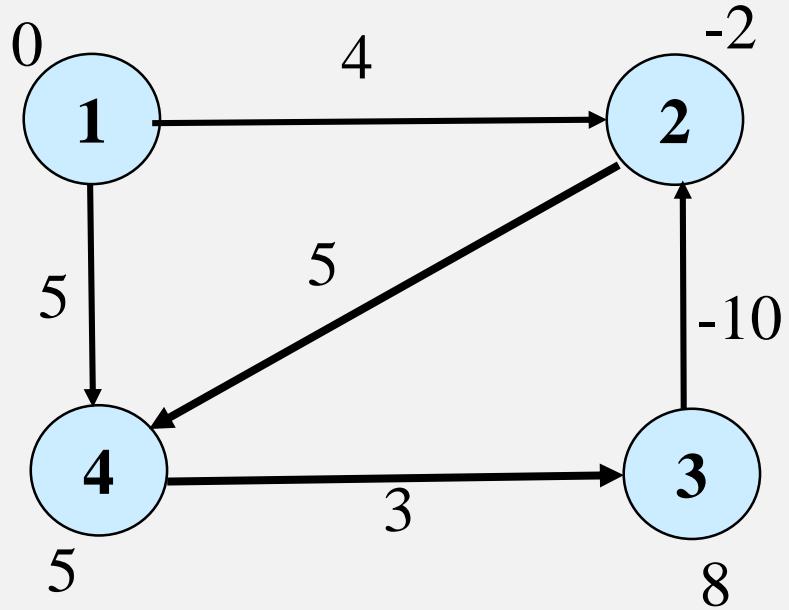
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-3

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

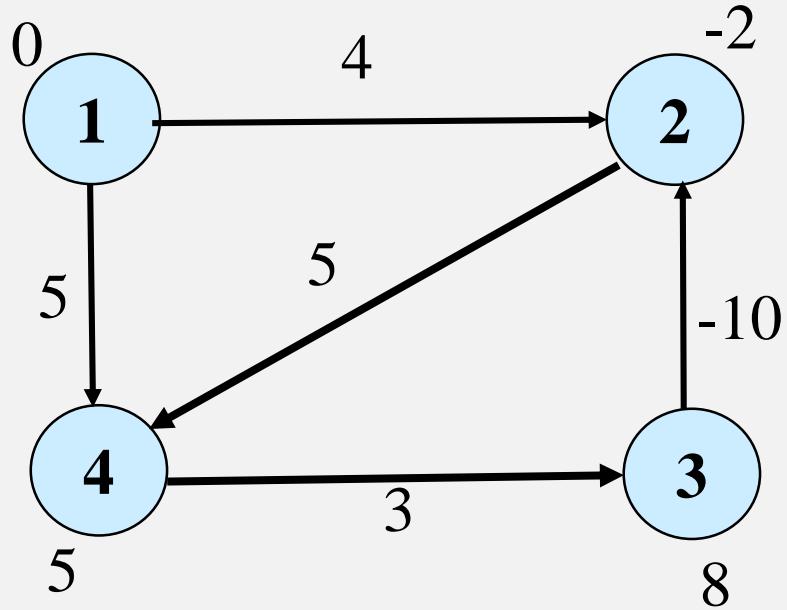
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-3

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

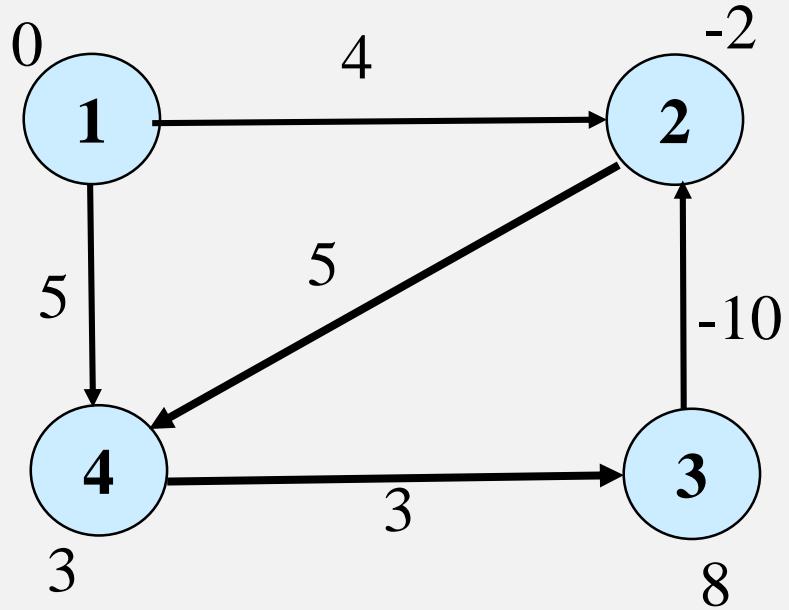
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-3

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

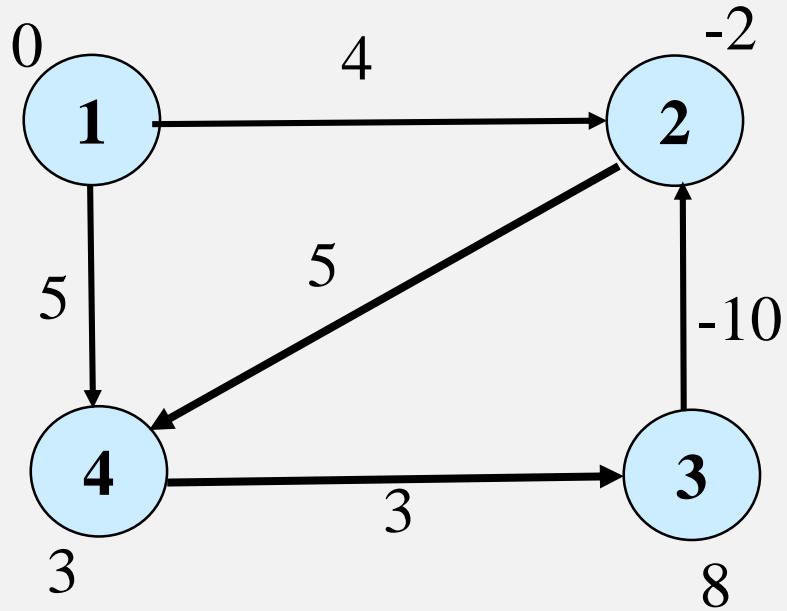
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-3

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

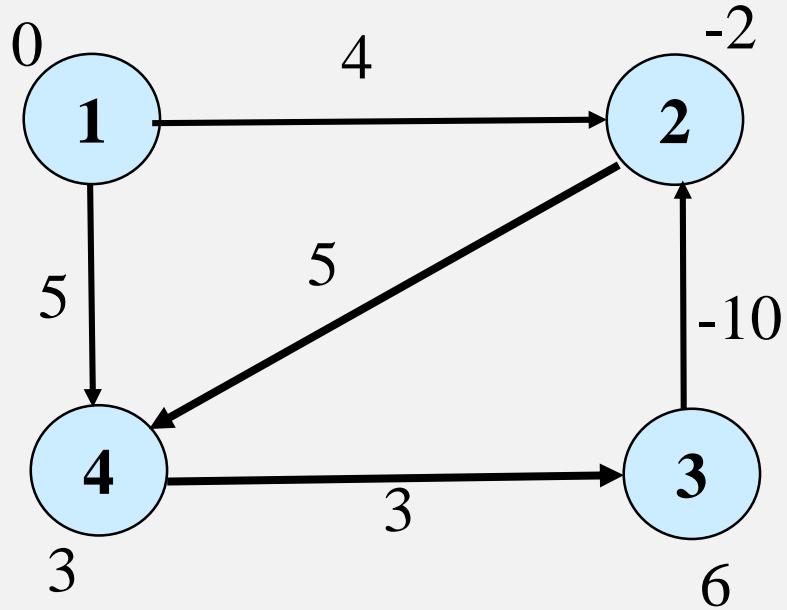
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-3

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

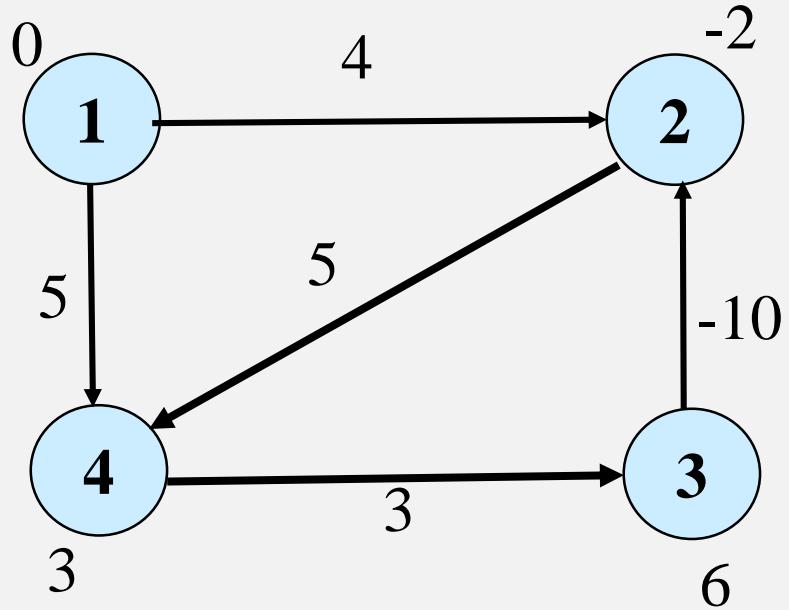
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
    
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
    
```

Example



Iteration-4

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

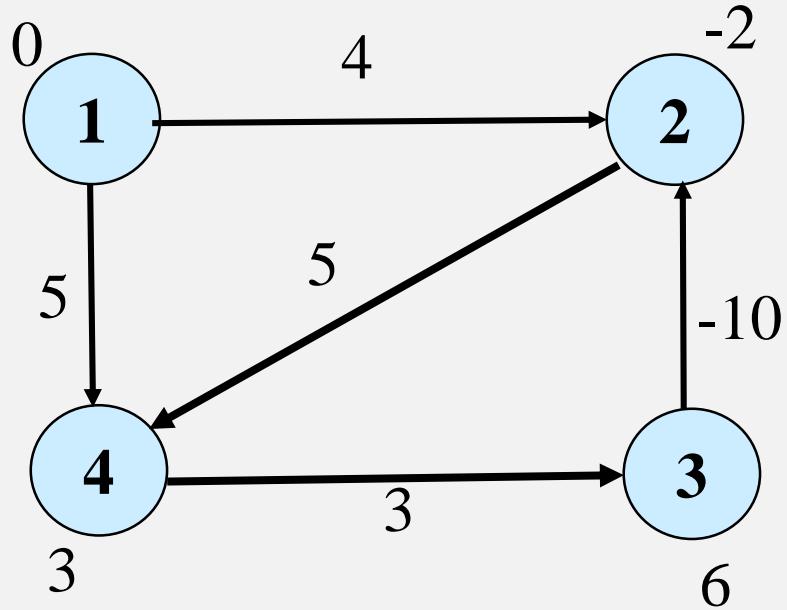
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
    
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
    
```

Example



Iteration-4

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

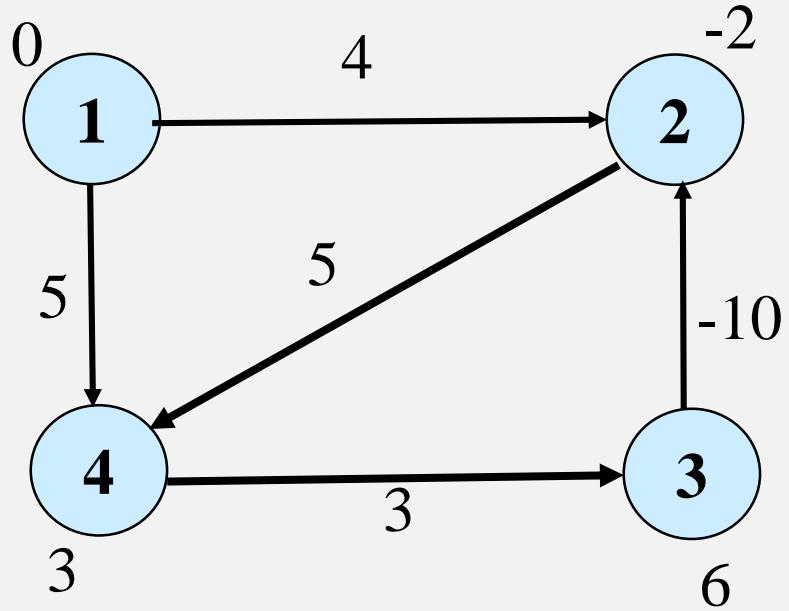
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
    
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
    
```

Example



Iteration-4

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

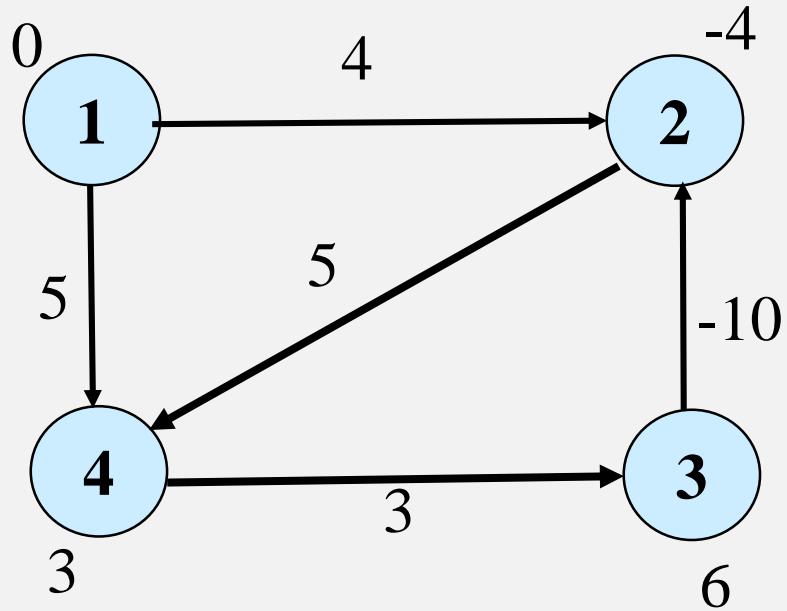
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-4

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

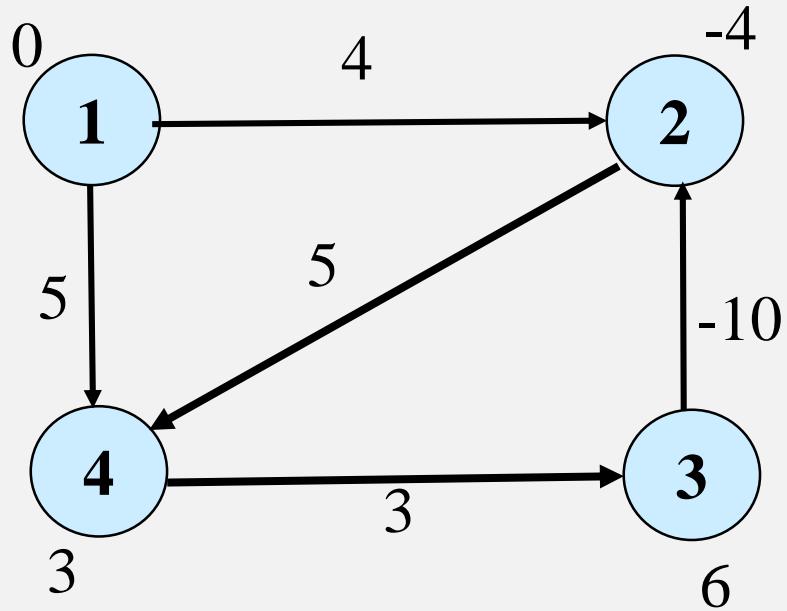
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-4

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

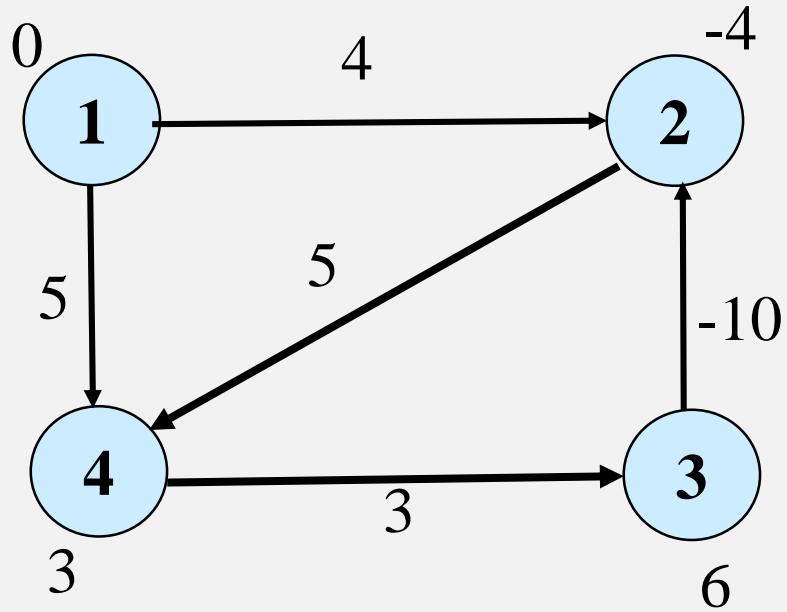
```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Example



Iteration-5

$$E(G)=\{(1,2), (1,4), (2,4), (3,2), (4,3)\}$$

```

Initialize(G, s);
for i := 1 to |V[G]| -1 do
    for each (u, v) in E[G] do
        Relax(u, v, w)
    end
end;
for each (u, v) in E[G] do
    if d[v] > d[u] + w(u, v) then
        return false
    end
end;
return true
  
```

```

Relax(u, v, w)
    if d[v] > d[u] + w(u, v) then
        d[v] := d[u] + w(u, v);
        π[v] := u
    end
  
```

Thank you