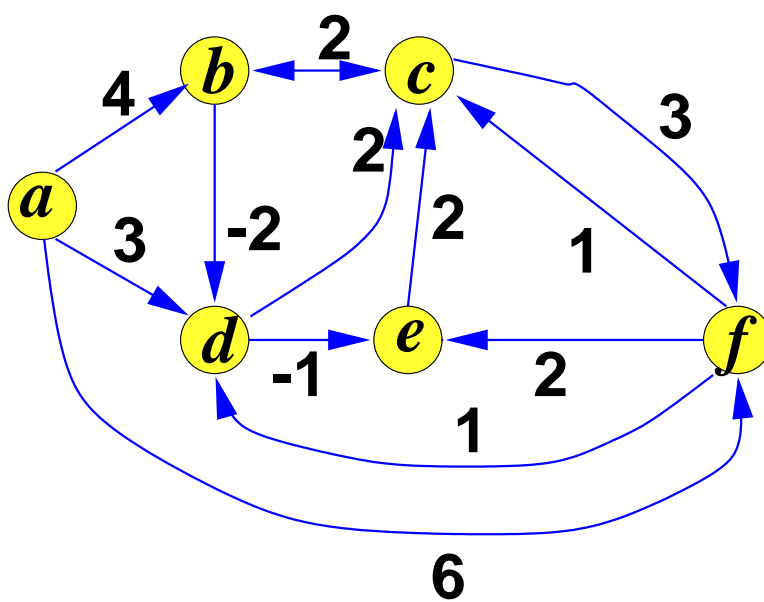


## Chapter 24: Single-Source Shortest Paths

The shortest path ... the path with the smallest edge-weight sum



What is the length of the path  $\langle a, b, d, c, f \rangle$ ?

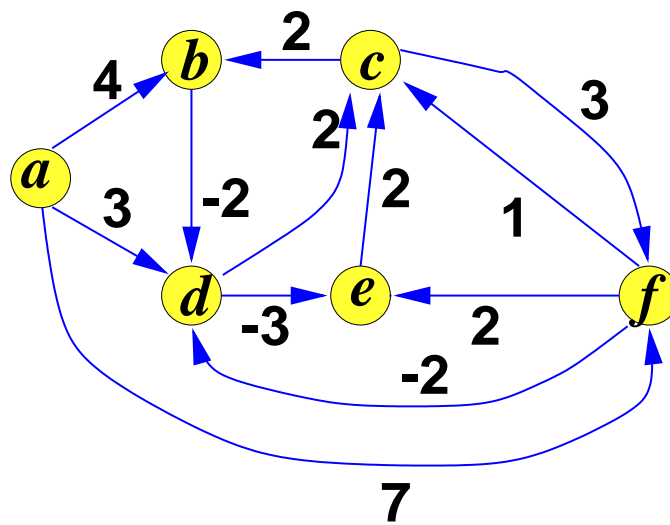
## The Shortest Path Problem (SPP)

$\delta(u, v) \stackrel{\text{def}}{=} \text{the shortest path length}$

Compute  $\delta(u, v)$  for:

1. **Single-Source:** a fixed  $u$  and all  $v$ ;
2. **Single-Destination:** a fixed  $v$  and all  $u$ ;
3. **Single-Pair:** fixed  $u$  and  $v$ ;
4. **All-Pair:** all  $u$  and  $v$ .

Negative weight edges can create negative weight cycles, which make the shortest paths undefined



## Theorem A

### Optimal Substructure Theorem

If  $p = \langle v_1, v_2, \dots, v_k \rangle$  is a shortest path from  $v_1$  to  $v_k$ , then for all  $i, j, 1 \leq i \leq j \leq k$ ,  $p_{ij} = \langle v_i, \dots, v_j \rangle$  is a shortest path from  $v_i$  to  $v_j$ .

1.  $\delta(u, v) = \min\{\delta(u, a) + w(a, v) \mid (a, v) \in E\}$ .
2.  $\delta(u, v) \leq \delta(u, a) + w(a, v)$  for any  $a$ .

**Idea** Use a variable  $d[u, v]$  to compute  $\delta(u, v)$

**Relaxation** with respect to points  $a, b, c$ :

$$d[a, c] \leftarrow \min\{d[a, c], d[a, b] + w(b, c)\}.$$

### A general strategy

1. Set  $d[u, v] = \infty$  for all  $u$  and  $v$ .
2. Repeat the following until **no update**
  - (\*) **Pick**  $a, b, c$  and, **relax**  $d[a, c]$  with  $(b, c)$ :
$$d[a, c] \leftarrow \min\{d[a, c], d[a, b] + w(b, c)\}.$$
3. Output  $d[u, v]$  as  $\delta(u, v)$ .

## Strategy for SSSP

Fix  $u$  to the source node  $s$ . Omit the first component from  $d$  since it's fixed to  $s$ .

1. Set  $d[v] = \infty$  for all  $v$ .
2. Repeat the following until no possibility of update:
  - (\*) Pick  $b$  and  $c$  and then relax  $d[c]$  with  $(b, c)$ :

$$d[c] \leftarrow \min\{d[c], d[b] + w(b, c)\}.$$

3. Output  $d[v]$  as  $\delta(s, v)$ .

## Dijkstra's Algorithm

Here all weights are nonnegative.

$S \stackrel{\text{def}}{=} \text{the set of nodes that are "finished," i.e.,}$   
 $\delta(s, v) = d[v]$   
 $Q \stackrel{\text{def}}{=} V - S, \text{ a priority queue keyed with } d$

1. Set  $d[s]$  to 0,  $d[v]$  to  $\infty$  for all  $v \neq s$ ,  $S$  to  $\emptyset$ , and  $Q$  to  $V$ .
2. While  $Q \neq \emptyset$ , "extract-min"  $u \in Q$ , then
  - (a) Add  $u$  to  $S$ .
  - (b) For each  $v$  with  $(u, v) \in E$ ,

$$d[v] \leftarrow \min\{d[v], d[u] + w(u, v)\}.$$

- ★ Each time  $d[v]$  is updated, set  $\pi[v]$  to  $u$ , which is the **predecessor** of  $v$ .

*How many times is each edge  
examined?*

*How many calls to  
**Extract-Min?***

*How about **Decrease-Key?***

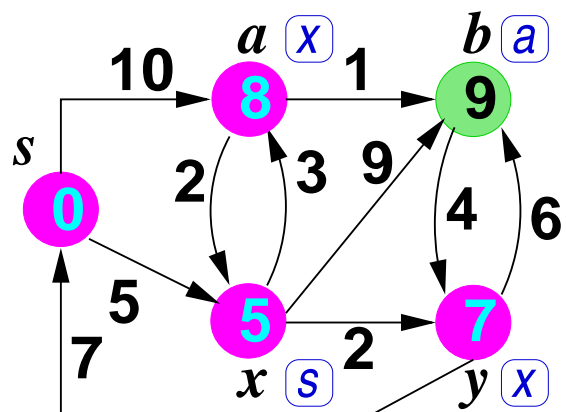
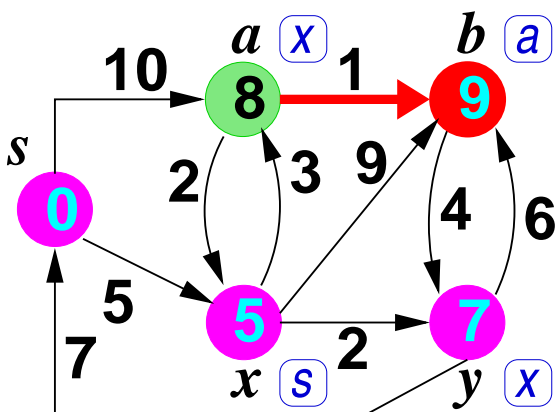
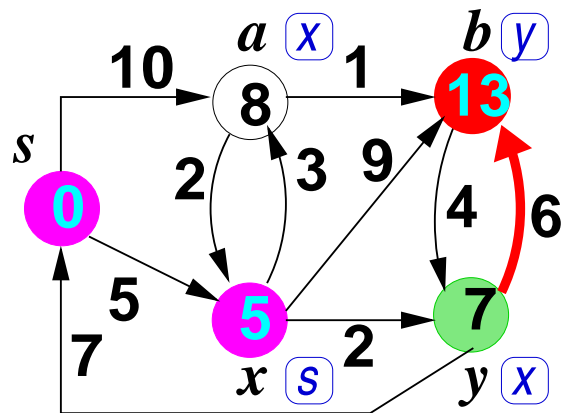
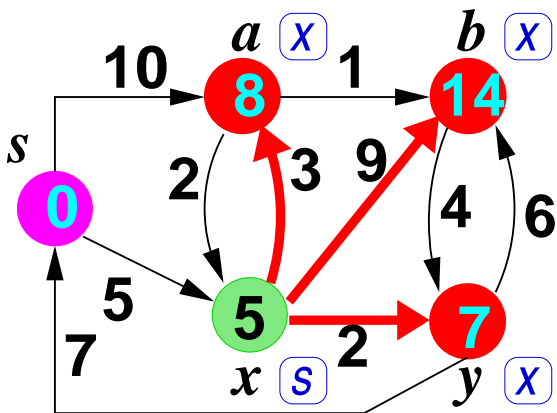
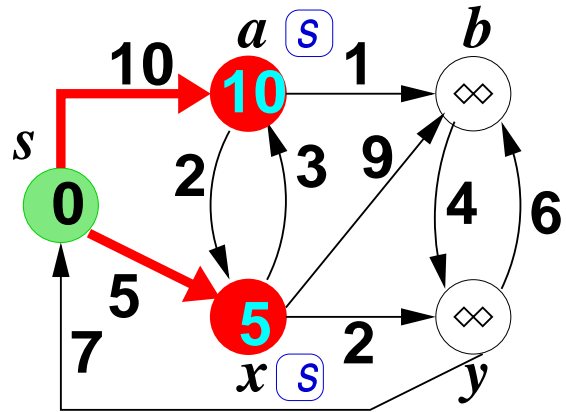
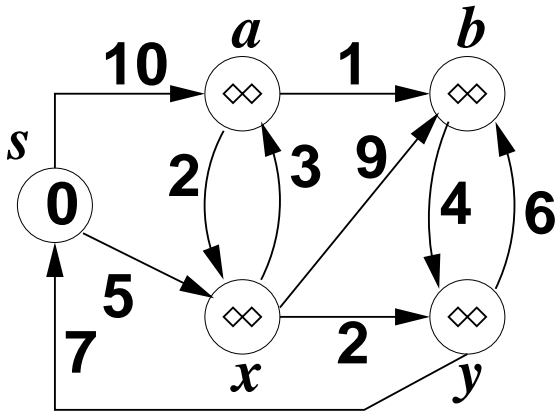
Each edge is examined once.

**Extract-Min** is called at most  $V$  times.

**Decrease-Key** is called at most  $E$  times.

With Fibonacci heaps, the total running time is  $O(E + V \lg V)$ .





## The Bellman-Ford Algorithm

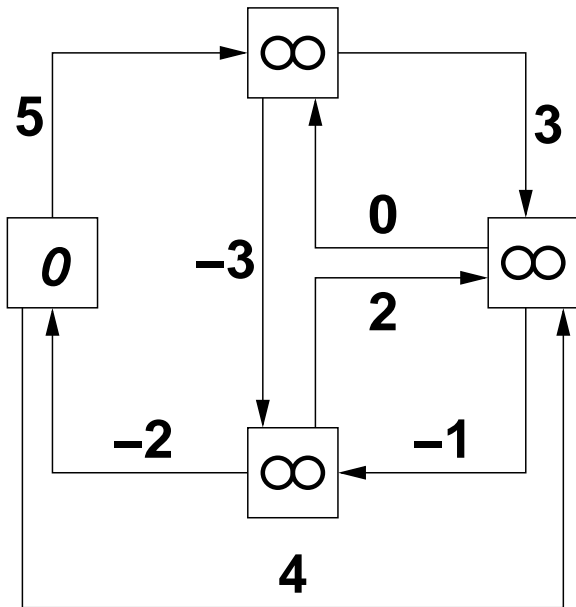
Here negative weights are allowed. The algorithm detects whether there are negative weight cycles.

Repeat  $V - 1$  times:

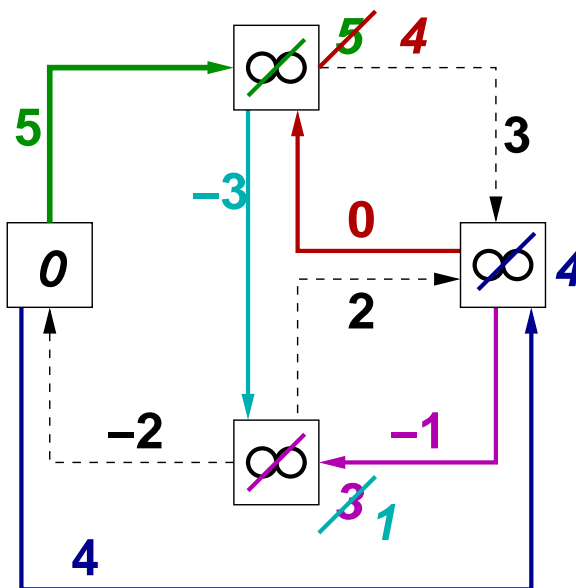
1. For each edge  $(u, v)$ , **relax with respect to  $(u, v)$**
2. If for some  $(u, v)$ ,  $d[v] > d[u] + w(u, v)$ , then output “negative weight cycles”

*How many times is each edge examined?*

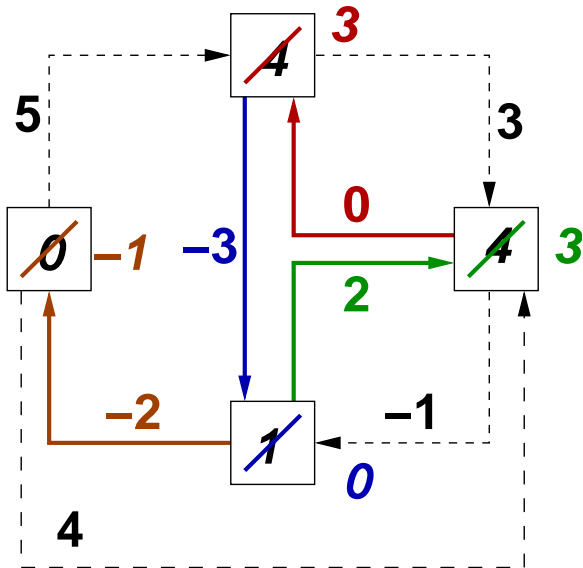
*What is the running time?*



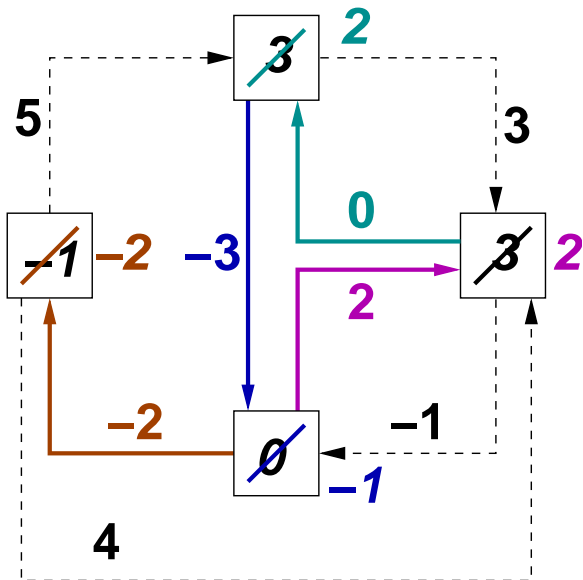
The input graph.  
The edges are ordered in decreasing order of their weights.



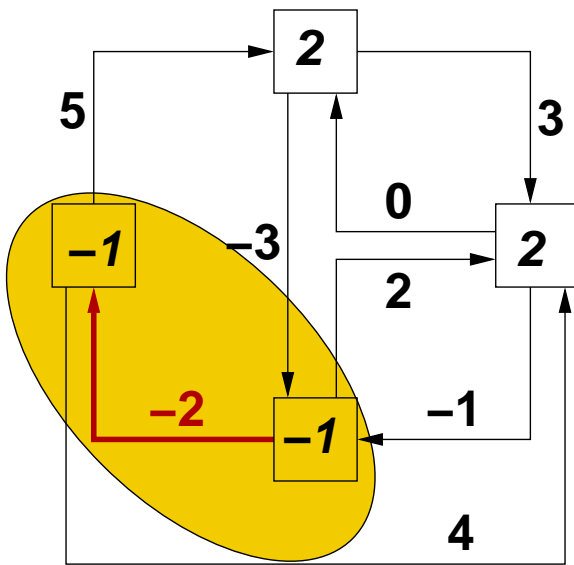
The first round.



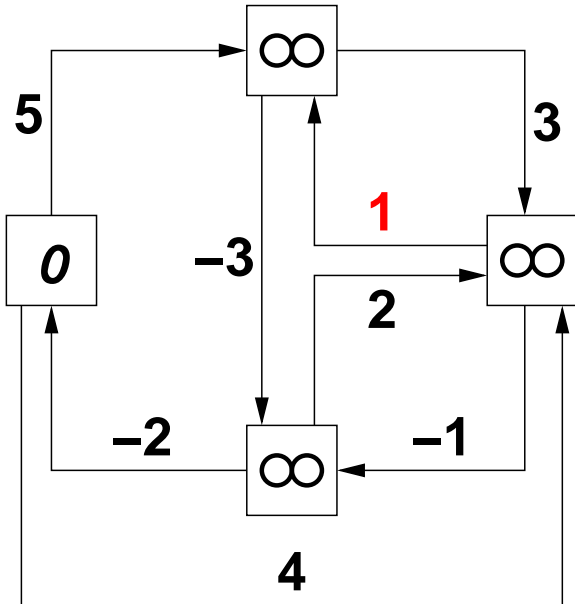
The second round.



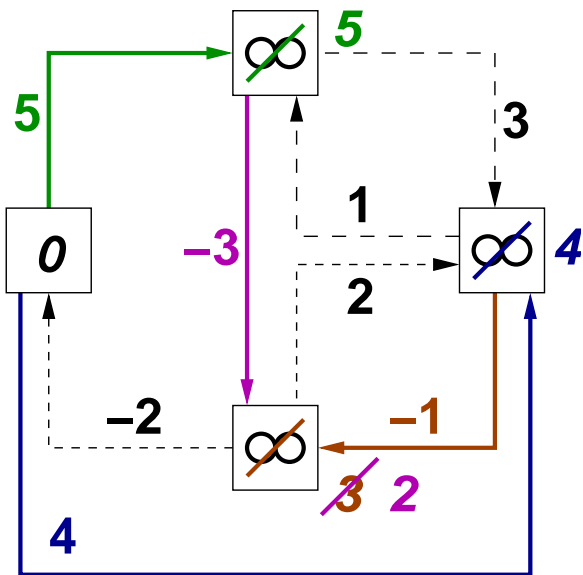
The third round.



Relaxation is still possible.



Another input graph. The edges are ordered in decreasing order of their weights.



The first round.

$\mu(u)$  = the minimum number of edges in shortest  $s$ - $u$  paths;  $\mu(u) \leq V - 1$  for all  $u$ .

**Theorem B** If  $G$  has no negative weight cycles then for all  $v$  after the  $\mu(v)$ -th round  $d[v]$  becomes  $\delta(s, v)$ .

**Proof** By induction on  $\mu(v)$ .

**(Base)**  $\mu(v) = 0$ : trivial because  $v = s$ .

**(Induction)** Let  $m \geq 0$  and  $\mu(v) = m + 1$ . Suppose that the claim holds for all  $u$  such that  $\mu(u) \leq m$ . Pick an  $(m + 1)$ -edge shortest  $s$ - $v$  path  $p$ , and let  $u$  be the node preceding  $v$ . Then  $\mu(u) = m$  and

$$\delta(s, v) = \delta(s, u) + w(u, v).$$

So after the  $m$ -th round  $d[v] = \delta(s, v)$ . Thus, in the  $(m + 1)$ st round,  $d[u]$  becomes **at most**  $\delta(s, u) + w(u, v)$ , and this is the smallest possible value that  $d[u]$  can acquire. ■



**Theorem C** If  $G$  has a negative weight cycle then there exist some  $u$  and  $v$  such that after the  $(V - 1)$ -st round  $d[v] > d[u] + w(u, v)$  holds. Thus, if  $G$  has a negative weight cycle then the algorithm outputs “negative weight cycle.”

**Proof** Suppose  $G$  has a negative weight cycle  $\langle v_1, v_2, \dots, v_k, v_1 \rangle$  whose length is  $L < 0$ . Then

$$L = w(v_1, v_2) + \dots + w(v_{k-1}, v_k) + w(v_k, v_1).$$

Assume, to the contrary, that  $d[v] \leq d[u] + w(u, v)$  holds for all  $u$  and  $v$  after the  $(V - 1)$ -st round. By our assumption,

$$d[v_i] \leq d[v_{i-1}] + w(v_{i-1}, v_i)$$

for all  $i, 1 \leq i \leq k$ , where  $v_0 = v_k$ . Summing these inequalities for all  $i$ , we have

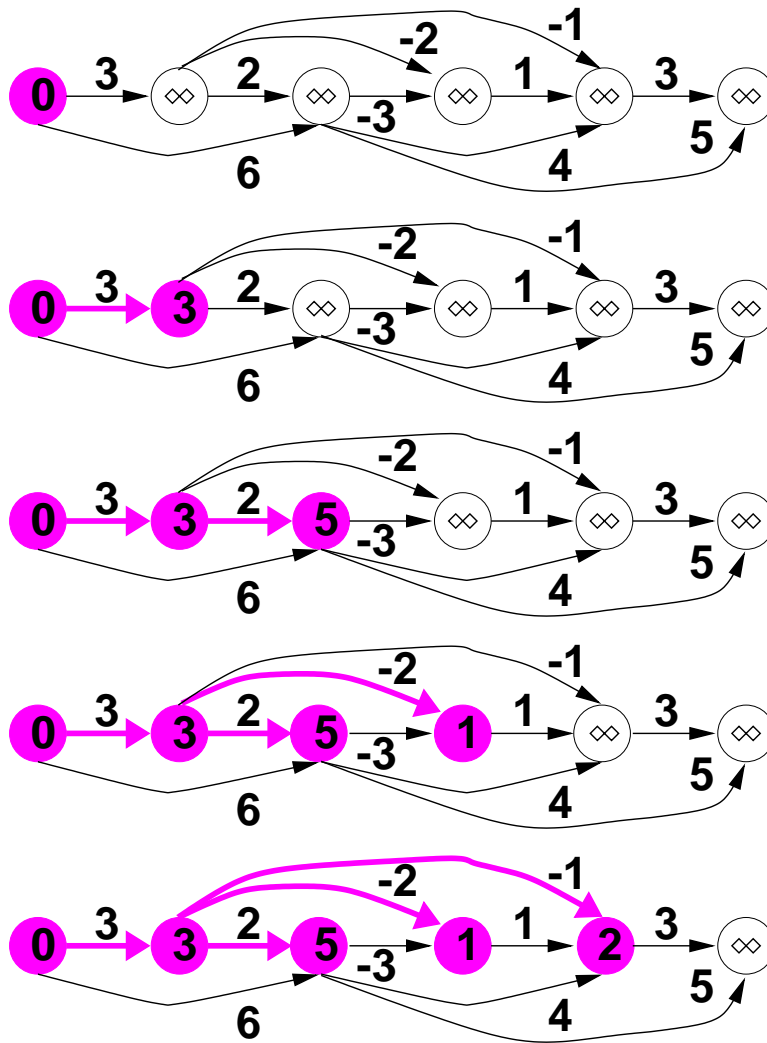
$$\sum_{i=1}^k d[v_i] \leq \sum_{i=1}^k d[v_i] + L.$$

This implies  $L \geq 0$ , a contradiction. ■

## SSSP for a DAG

1. Obtain a topological sort of the nodes.
2. For each  $u \neq s$ , set  $d[u] = \infty$ . Set  $d[s] = 0$ .
3. For each node  $v$  in the sorted order, and each  $u$  with  $(u, v) \in E$ , set  $d[v] = \min\{d[v], d[u] + w(u, v)\}$ .

*What is the running time?*



*What is the value of the last node?*