3-7 Relax! We are SSSP Algorithms.

Hengfeng Wei

hfwei@nju.edu.cn

November 12, 2018

Definition (Shortest Path)

$$G = (V, E, w) : \text{ weighted digraph}$$

$$\delta(u, v) = \begin{cases} \min \{ w(p) : u \overset{p}{\rightsquigarrow} v \} & \text{if } u \rightsquigarrow v \\ \infty & \text{o.w.} \end{cases}$$

Definition (Shortest Path)

$$G = (V, E, w) : \text{ weighted digraph}$$

$$\delta(u, v) = \begin{cases} \min \{ w(p) : u \overset{p}{\rightsquigarrow} v \} & \text{if } u \rightsquigarrow v \\ \infty & \text{o.w.} \end{cases}$$

Path *vs.* Simple path

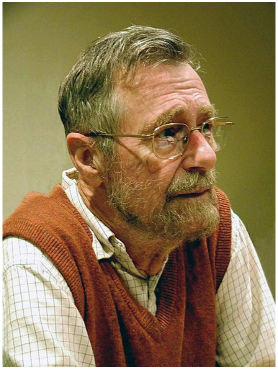
Definition (Shortest Path)

$$G = (V, E, w) : \text{ weighted digraph}$$

$$\delta(u, v) = \begin{cases} \min \{ w(p) : u \overset{p}{\rightsquigarrow} v \} & \text{if } u \rightsquigarrow v \\ \infty & \text{o.w.} \end{cases}$$

Path *vs.* Simple path

Q : What about undirected graphs?



For fundamental contributions to programming as a high, intellectual challenge;

for eloquent insistence and practical demonstration that programs should be composed correctly, not just debugged into correctness;

for illuminating perception of problems at the foundations of program design.

— *Turing Award*, 1972

```
1: procedure DIJKSTRA($G, w, s$)
2:     INIT-SINGLE-SOURCE($G, s$)
3:     $S = \emptyset$
4:     $Q = G.V$
5:     while $Q \neq \emptyset$ do
6:         $u \leftarrow$ EXTRACT-MIN($Q$)
7:         $S \leftarrow S \cup \{u\}$
8:         for $v \in G.Adj[u]$ do
9:             RELAX($u, v, w$)
```

```
1: procedure DIJKSTRA($(G, w, s)$)
2:     INIT-SINGLE-SOURCE($(G, s)$)
3:     $S = \emptyset$
4:     $Q = G.V$
5:     while $Q \neq \emptyset$ do
6:         $u \leftarrow$ EXTRACT-MIN($(Q)$)
7:         $S \leftarrow S \cup \{u\}$
8:         for $v \in G.Adj[u]$ do
9:             RELAX($(u, v, w)$)
```

Array: $O(n^2)$

Min-heap: $O(E \log V)$

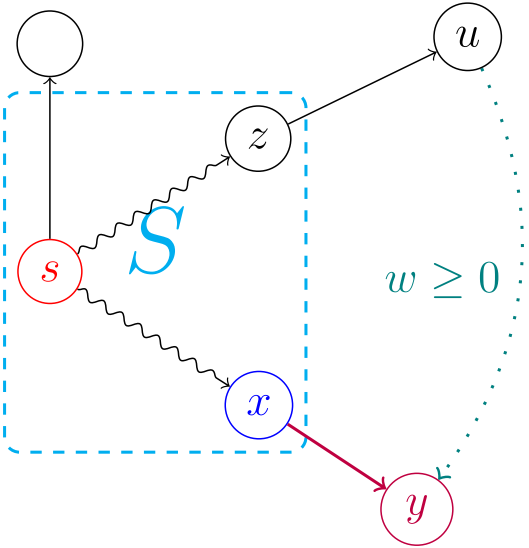
Fib-heap: $O(V \log V + E)$

```
1: procedure DIJKSTRA($G, w, s$)
2:     INIT-SINGLE-SOURCE($G, s$)
3:     $S = \emptyset$
4:     $Q = G.V$
5:     while $Q \neq \emptyset$ do
6:         $u \leftarrow$ EXTRACT-MIN($Q$)
7:         $S \leftarrow S \cup \{u\}$
8:         for $v \in G.Adj[u]$ do
9:             RELAX($u, v, w$)
```

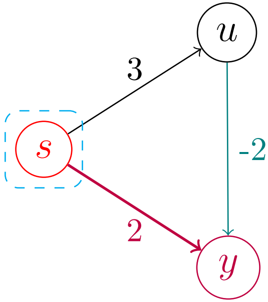
Array: $O(n^2)$

Min-heap: $O(E \log V)$

Fib-heap: $O(V \log V + E)$


u
z
S
s
$w \geq 0$
x
y

Negative-weight Edges for Dijkstra's Algorithm (Problem 24.3-2)

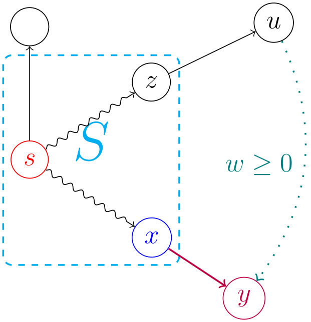


Negative-weight Edges for Dijkstra's Algorithm (Additional Problem 24.3-10)

- All negative-weight egdes are from s
- No negative-weight cycles

Negative-weight Edges for Dijkstra's Algorithm (Additional Problem 24.3-10)

- All negative-weight egdes are from s
- No negative-weight cycles



Checking Output of Dijkstra's Algorithm (Problem 24.3-4)

Dijkstra’s Algorithm on Digraph with Nonnegative-weight Edges

Lawler's Algorithm on DAG

$\Longleftarrow$

Dijkstra's Algorithm on Digraph with Nonnegative-weight Edges

$\Longrightarrow$

Bellman-Ford Algorithm on Digraph with Negative-weight Edges

```
procedure DAG-SSSP($G, w, s$)
    INIT-SINGLE-SOURCE($G, s$)
    TOPO-SORT($G$)
    for $u \in V$ in topo. order do
        for $v \in G.Adj[u]$ do
            RELAX($u, v, w$)
```

```
1: procedure DAG-SSSP($G, w, s$)
2:     INIT-SINGLE-SOURCE($G, s$)
3:     TOPO-SORT($G$)
4:     for $u \in V$ in topo. order do
5:         for $v \in G.Adj[u]$ do
6:             RELAX($u, v, w$)
```

$$\Theta(V + E)$$

```
1: procedure DAG-SSSP($G, w, s$)
2:     Init-Single-Source($G, s$)
3:     Topo-Sort($G$)
4:     for $u \in V$ in topo. order do
5:         for $v \in G.Adj[u]$ do
6:             Relax($u, v, w$)
```

```
1: procedure Dijkstra($G, w, s$)
2:     Init-Single-Source($G, s$)
3:     $Q = G.V$
4:     while $Q \neq \emptyset$ do
5:         $u \leftarrow$ Extract-Min($Q$)
6:         for $v \in G.Adj[u]$ do
7:             Relax($u, v, w$)
```

```
procedure DAG-SSSP($G, w, s$)
    Init-Single-Source($G, s$)
    Topo-Sort($G$)
    for $u \in V$ in topo. order do
        for $v \in G.Adj[u]$ do
            Relax($u, v, w$)
```

```
procedure Dijkstra($G, w, s$)
    Init-Single-Source($G, s$)
    $Q = G.V$
    while $Q \neq \emptyset$ do
        $u \leftarrow$ Extract-Min($Q$)
        for $v \in G.Adj[u]$ do
            Relax($u, v, w$)
```

```
1: procedure DAG-SSSP($(G, w, s)$)
2:     INIT-SINGLE-SOURCE($(G, s)$)
3:     TOPO-SORT($(G)$)
4:     for $u \in V$ in topo. order do
5:         for $v \in G.Adj[u]$ do
6:             RELAX($(u, v, w)$)
```

```
1: procedure DIJKSTRA($(G, w, s)$)
2:     INIT-SINGLE-SOURCE($(G, s)$)
3:     $Q = G.V$
4:     while $Q \neq \emptyset$ do
5:         $u \leftarrow$ EXTRACT-MIN($(Q)$)
6:         for $v \in G.Adj[u]$ do
7:             RELAX($(u, v, w)$)
```

Q_1 : Why is $\delta(s, u)$ determined right now?

```
1: procedure DAG-SSSP($G, w, s$)
2:     INIT-SINGLE-SOURCE($G, s$)
3:     TOPO-SORT($G$)
4:     for $u \in V$ in topo. order do
5:         for $v \in G.Adj[u]$ do
6:             RELAX($u, v, w$)
```

```
1: procedure DIJKSTRA($G, w, s$)
2:     INIT-SINGLE-SOURCE($G, s$)
3:     $Q = G.V$
4:     while $Q \neq \emptyset$ do
5:         $u \leftarrow$ EXTRACT-MIN($Q$)
6:         for $v \in G.Adj[u]$ do
7:             RELAX($u, v, w$)
```

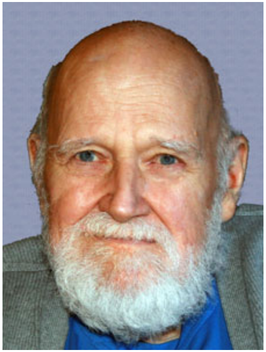
Q_1 : Why is $\delta(s, u)$ determined right now?

Q_2 : Is $\delta(s, u)$ determined in non-decreasing order?

Little Modification to DAG-SSSP (Problem 24.2-2)

```
1: procedure DAG-SSSP($G, w, s$)
2:     INIT-SINGLE-SOURCE($G, s$)
3:     TOPO-SORT($G$)
4:     for the first $|V| - 1$ vertices $u \in V$ in topo. order do
5:         for $v \in G.Adj[u]$ do
6:             RELAX($u, v, w$)
```

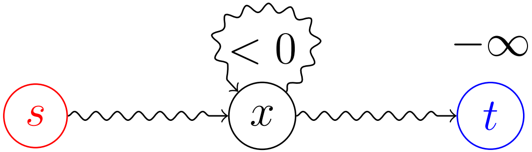
Richard Bellman (1920—1984)



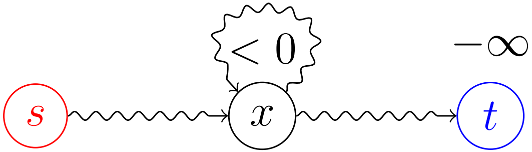
Lester Randolph Ford Jr. (1927—2017)

```
procedure BELLMAN-FORD($G, w, s$)
    INIT-SINGLE-SOURCE($G, s$)

    for $i \leftarrow 1$ to $|V| - 1$ do
        for $(u, v) \in E$ do
            RELAX($u, v, w$)

    for $(u, v) \in E$ do
        if $v.d > u.d + w(u, v)$ then
            return FALSE

    return TRUE
```

Deal with Negative-weight Cycles (Problem 24.1-4)

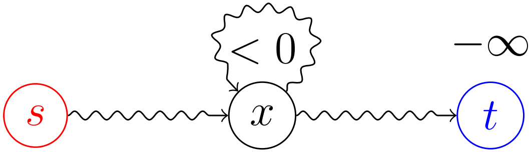


Deal with Negative-weight Cycles (Problem 24.1-4)



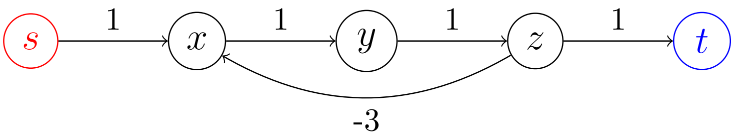
```
1: procedure Bellman-Ford-NC-Wrong($(G, w, s)$)
2:     Init-Single-Source($(G, s)$)
3:     for $i \leftarrow 1$ to $|V| - 1$ do
4:         for $(u, v) \in E$ do
5:             Relax($(u, v, w)$)
6:     for $(u, v) \in E$ do
7:         if $v.d > u.d + w(u, v)$ then
8:             $v.d = -\infty$
```

Deal with Negative-weight Cycles (Problem 24.1-4)

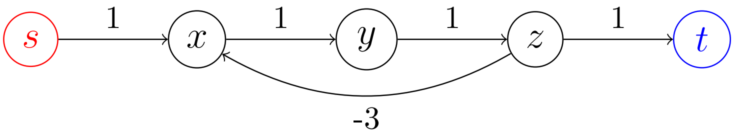


```
procedure BELLMAN-FORD-NC-WRONG($G, w, s$)
    INIT-SINGLE-SOURCE($G, s$)
    for $i \leftarrow 1$ to $|V| - 1$ do
        for $(u, v) \in E$ do
            RELAX($u, v, w$)
    for $(u, v) \in E$ do
        if $v.d > u.d + w(u, v)$ then
            $v.d = -\infty$
```

$$\begin{tikzcd}
s \arrow[r, "1"] & x \arrow[r, "1"] & y \arrow[r, "1"] & z \arrow[r, "1"] \arrow[ll, bend left, "-3"] & t
\end{tikzcd}$$



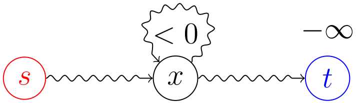
```
procedure BELLMAN-FORD-NC-WRONG($G, w, s$)
    INIT-SINGLE-SOURCE($G, s$)

    for $i \leftarrow 1$ to $|V| - 1$ do
        for $(u, v) \in E$ do
            RELAX($u, v, w$)

    for $i \leftarrow 1$ to $|V| - 1$ do                    $\triangleright$
        for $(u, v) \in E$ do
            if $v.d > u.d + w(u, v)$ then
                $v.d = -\infty$
```



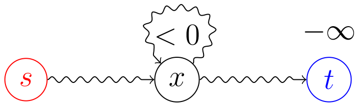
```
1: procedure BELLMAN-FORD-NC-WRONG($(G, w, s)$)
2:     INIT-SINGLE-SOURCE($(G, s)$)
3:     for $i \leftarrow 1$ to $|V| - 1$ do
4:         for $(u, v) \in E$ do
5:             RELAX($(u, v, w)$)
6:     for $i \leftarrow 1$ to $|V| - 1$ do                ▷ $\Theta(VE)$
7:         for $(u, v) \in E$ do
8:             if $v.d > u.d + w(u, v)$ then
9:                 $v.d = -\infty$
```

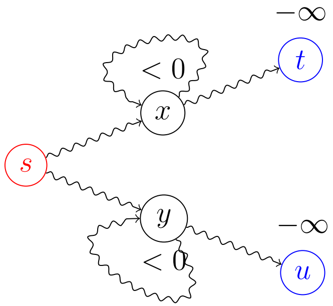
$$O(V + E)$$

$$O(V + E)$$

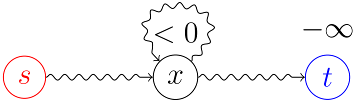


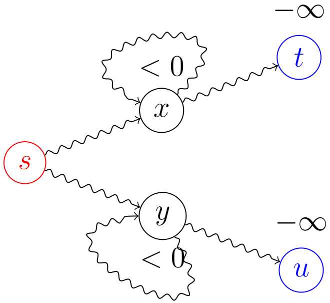
$$O(V + E)$$





$O(V + E)$





Theorem (The $|V|$-th Pass of Bellman-Ford Algorithm)

For every reachable negative-weight cycle, at least one edge of it has been relaxed in the $|V|$-th pass.

Terminate Early in Bellman-Ford Algorithm (Problem 24.1-3)

$G = (V, E)$ without negative-weight cycles

$$m \triangleq \min_{v \in V} \Big\{ \text{Len}\big(\delta(s, v)\big) \Big\} \text{ (Unknown!)}$$

Terminate Early in Bellman-Ford Algorithm (Problem 24.1-3)

$$G = (V, E) \text{ without negative-weight cycles}$$

$$m \triangleq \min_{v \in V} \left\{ \text{Len}(\delta(s, v)) \right\} \text{ (Unknown!)}$$

```
procedure BELLMAN-FORD($G, w, s$)
    INIT-SINGLE-SOURCE($G, s$)

    $f \leftarrow$ FALSE
    for $i \leftarrow 1$ to $|V| - 1$ do
        for $(u, v) \in E$ do
            if $v.d > u.d + w(u, v)$ then
                $v.d = u.d + w(u, v)$
                $f \leftarrow$ TRUE

        if $f =$ FALSE then
            return
```

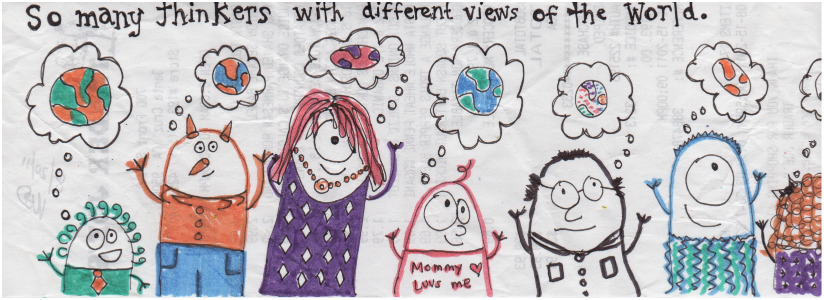
Terminate Early in Bellman-Ford Algorithm (Problem 24.1-3)

$$G = (V, E) \text{ without negative-weight cycles}$$

$$m \triangleq \min_{v \in V} \left\{ \text{Len}(\delta(s, v)) \right\} \text{ (Unknown!)}$$

```
1:  procedure BELLMAN-FORD($(G, w, s)$)
2:      INIT-SINGLE-SOURCE($(G, s)$)
3:      $f \leftarrow$ FALSE
4:      for $i \leftarrow 1$ to $|V| - 1$ do
5:          for $(u, v) \in E$ do
6:              if $v.d > u.d + w(u, v)$ then
7:                  $v.d = u.d + w(u, v)$
8:                  $f \leftarrow$ TRUE
9:          if $f =$ FALSE then
10:             return
```

Two Different Views of Bellman-Ford Algorithms:



```
1: procedure DIJKSTRA($(G, w, s)$)
2:     INIT-SINGLE-SOURCE($(G, s)$)
3:     $Q = G.V$
4:     while $Q \neq \emptyset$ do
5:         $u \leftarrow$ EXTRACT-MIN($(Q)$)
6:         for $v \in G.Adj[u]$ do
7:             RELAX($(u, v, w)$)
```

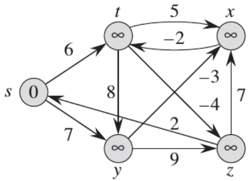
```
1: procedure BELLMAN-FORD($(G, w, s$
2:     INIT-SINGLE-SOURCE($(G, s)$)
3:     for $i \leftarrow 1$ to $|V| - 1$ do
4:         for $(u, v) \in E$ do
5:             RELAX($(u, v, w)$)
6:     for $(u, v) \in E$ do
7:         if $v.d > u.d + w(u, v)$ then
8:             return FALSE
9:     return TRUE
```

```
1: procedure Dijkstra$(G, w, s)$
2:     Init-Single-Source$(G, s)$
3:     $Q = G.V$
4:     while $Q \neq \emptyset$ do
5:         $u \leftarrow$ Extract-Min$(Q)$
6:         for $v \in G.Adj[u]$ do
7:             Relax$(u, v, w)$
```

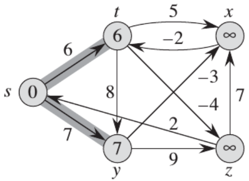
```
1: procedure Bellman-Ford$(G, w, s$
2:     Init-Single-Source$(G, s)$
3:     for $i \leftarrow 1$ to $|V| - 1$ do
4:         for $(u, v) \in E$ do
5:             Relax$(u, v, w)$
6:     for $(u, v) \in E$ do
7:         if $v.d > u.d + w(u, v)$ then
8:             return False
9:     return True
```

$\boxed{\text{Bellman-Ford Algorithm} \equiv \text{Dijkstra's Algorithm with Queue}}$

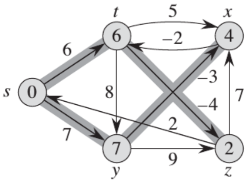
Bellman-Ford Algorithm $\equiv$ Dijkstra's Algorithm with Queue



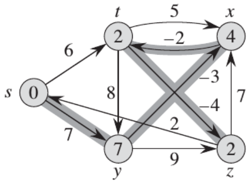
(a)



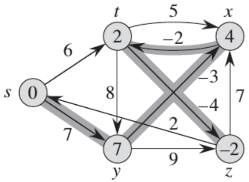
(b)



(c)



(d)



(e)

$$\boxed{\text{Bellman-Ford Algorithm is a DP Algorithm.}}$$

$\boxed{\text{Bellman-Ford Algorithm is a DP Algorithm.}}$

$d[i, v]$: the length of the shortest path $s \rightsquigarrow v$ consisting of $\leq i$ edges

$$\boxed{\text{Bellman-Ford Algorithm is a DP Algorithm.}}$$

$d[i,v]$: the length of the shortest path $s \rightsquigarrow v$ consisting of $\le i$ edges

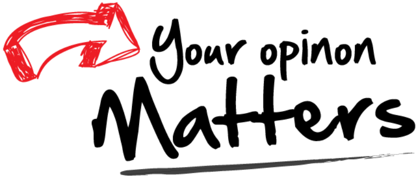
$$d[i,v] = \begin{cases} 0 & i = 0 \land v = s \\ \infty & i = 0 \land v \neq s \\ \min \Big\{ d[i-1,v], \min\limits_{(u,v) \in E} \{ d[i-1,u] + w(u,v) \} \Big\} & \text{o.w.} \end{cases}$$

```
procedure $\textsc{Bellman-Ford-DP}(G, w, s)$
    $d[0, s] \leftarrow 0$
    for $(v \neq s) \in V$ do
        $d[0, v] \leftarrow \infty$

    for $i \leftarrow 1$ to $|V| - 1$ do
        for $v \in V$ do
            $d[i, v] = d[i - 1, v]$
            for $(u, v) \in E$ do
                if $d[i - 1, v] > d[i - 1, u] + w(u, v)$ then
                    $d[i, v] = d[i - 1, u] + w(u, v)$
```

```
procedure BELLMAN-FORD-DP($(G, w, s)$)
    $d[0, s] \leftarrow 0$
    for $(v \neq s) \in V$ do
        $d[0, v] \leftarrow \infty$

    for $i \leftarrow 1$ to $|V| - 1$ do
        for $v \in V$ do
            $d[i, v] = d[i - 1, v]$
            for $(u, v) \in E$ do
                if $d[i - 1, v] > d[i - 1, u] + w(u, v)$ then
                    $d[i, v] = d[i - 1, u] + w(u, v)$
```

$$Q : d[i, v] \implies d[v]?$$

THANK YOU



Office 302

Mailbox: H016

hfwei@nju.edu.cn