

# 3-8 Cool? We are APSP Algorithms.

Hengfeng Wei

hfwei@nju.edu.cn

November 19, 2018





Please Help Me Out Here.

## Definition (Shortest Path)

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

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

Path *vs.* Simple path

Shortest-path Problem *vs.* Longest-path Problem

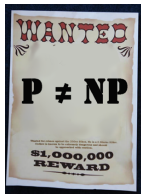
Digraph *vs.* Undirected Graph

## Single Source      Digraph

Shortest-path Problem    *vs.*    Longest-path Problem

$$\text{SP in } G \iff \text{LP in } -G$$

$O(VE)$  (Bellman-Ford)    *vs.*    NP-hard (I just told you.)



## Definition (Shortest Path)

$G = (V, E, w)$  : weighted digraph

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

$Q$  : How does Bellman-Ford Handle with Negative-weight Cycles?

$A$  : Report it and Treat Shortest Path as “Undefined”.

$$O(VE)$$

## Definition (Shortest Simple Path)

$G = (V, E, w)$  : weighted digraph

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

$Q$  : How Should an Algorithm for Shortest Simple Path Problem Handle with Negative-weight Cycles?

$A$  : Still to Find Shortest Simple Path.

NP-hard

Shortest Path Problem  $\iff$  Longest Path Problem

Shortest Simple Path Problem  $\iff$  Longest Simple Path Problem

Single Source      Undirected Graph

Negative-weight edges allowed    (Why?)

Simple path                      (Why?)

No negative-weight cycles    (o.w., NP-hard)

Single-source  $s \rightsquigarrow$  Single-target  $t$



## Shortest Path Algorithms

Luis Goddyn, Math 408

Given an edge weighted graph  $(G, d)$ ,  $d : E(G) \rightarrow \mathbb{Q}$  and two vertices  $s, t \in V(G)$ , the *Shortest Path Problem* is to find an  $s, t$ -path  $P$  whose total weight is as small as possible. Here,  $G$  may be either directed or undirected. A path in a graph is a sequence  $v_0 e_1 v_1 \dots v_k$  of vertices and edges such that no vertex or edge appears twice, and  $e_i$  joins  $v_{i-1}$  to  $v_i$ . If  $G$  is directed, then  $e_i$  should be oriented from  $v_{i-1}$  to  $v_i$ .

## Minimum-weight Perfect Matching

We leave it to the reader to  $\dots$

It is easy to check that  $\dots$

Why? This will be a homework question.

And Errors.

INTERESTED?  
*let's talk.*



## Robert W. Floyd (1936–1901)

*For having a clear influence on **methodologies** for the creation of efficient and reliable software, and for helping to **found** the following important subfields of computer science:*

*the theory of parsing, the semantics of programming languages, automatic program verification, automatic program synthesis, and analysis of algorithms*

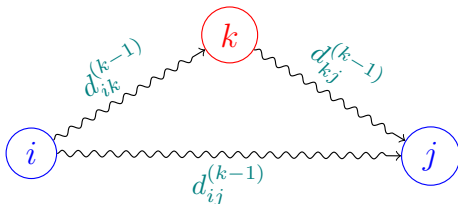
— *Turing Award*, 1978

$$d_{ij}^{(k)} :$$

the weight of a shortest path from  $i$  to  $j$   
for which all *intermediate* vertices are in  $\{1, 2, \dots, k\}$

$$D^{(n)} \triangleq \left( d_{ij}^{(n)} \right)$$

$k \in \text{SP}_{ij}^{(k)}?$



$$d_{ij}^{(k)} = \begin{cases} w_{ij} & k = 0 \\ \min \left\{ d_{ij}^{(k-1)}, \underbrace{d_{ik}^{(k-1)} + d_{kj}^{(k-1)}}_{\text{why?}} \right\} & k \geq 1 \end{cases}$$

$\dots$ , but we assume that there are **no** negative-weight cycles.

— Section 25.2 of CLRS

---

```
1: procedure FLOYD-WARSHALL( $W$ )
2:    $D^{(0)} = W$ 
3:   for  $k \leftarrow 1$  to  $n$  do
4:      $D^{(k)} \triangleq \left(d_{ij}^{(k)}\right) \leftarrow$  a new  $n \times n$  matrix
5:     for  $i \leftarrow 1$  to  $n$  do
6:       for  $j \leftarrow 1$  to  $n$  do
7:          $d_{ij}^{(k)} = \min \left\{ d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \right\}$ 
8:   return  $D^{(n)}$ 
```

---

$$\text{Space} : \Theta(n^3) \implies \Theta(n^2)$$

## FLOYD-WARSHALL Made Simple (Problem 25.2-4)

---

```
1: procedure FLOYD-WARSHALL-SIMPLIFIED( $W$ )
2:    $D = W$ 
3:   for  $k \leftarrow 1$  to  $n$  do
4:     for  $i \leftarrow 1$  to  $n$  do
5:       for  $j \leftarrow 1$  to  $n$  do
6:          $d_{ij} = \min \{d_{ij}, d_{ik} + d_{kj}\}$ 
7:   return  $D$ 
```

---

“Decrease” does no harm to the correctness.

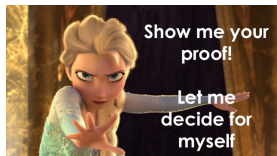
## Negative-weight Cycle Detection (Problem 25.2-6)

To detect negative-weight cycle (NC) using FLOYD-WARSHALL.

$$\exists i : d_{ii}^{(n)} < 0$$

Proof.

**The  
proof is  
trivial.**



$$\exists i : d_{ii}^{(n)} < 0 \iff \exists \text{ NC} \subseteq G$$

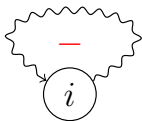




$$\exists i : d_{ii}^{(n)} < 0 \iff \exists \text{ NC} \subseteq G$$

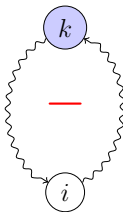
Proof.

“  $\implies$  ”



A Negative Cycle

“  $\Longleftarrow$  ”



A Simple Negative Cycle

$k : \max \#$

$$d_{ii}^{(k)} < 0$$



$l_{ij}^{(m)}$  : the length of a shortest path from  $i$  to  $j$  consisting of  $\leq m$  edges

$$l_{ij}^{(m)} = \min_{1 \leq k \leq n} \{l_{ik}^{(m-1)} + w_{kj}\}, \quad m \geq 2$$

$$l_{ij}^{(1)} = w_{ij}$$

$$L^{(n-1)} \triangleq \left( l_{ij}^{(n-1)} \right)$$

$$L^{(n-1)} = W^{n-1} \triangleq \underbrace{((W \cdot W) \cdot \dots)}_{n-1} \cdot W$$

## Associative EXTEND-SHORTEST-PATHS

Show that matrix multiplication defined by EXTEND-SHORTEST-PATHS is associative.

$$(W^a W^b) W^c = W^a (W^b W^c) \quad \text{vs.} \quad (WW)W = W(WW)$$

$(i, j)$

$$\begin{aligned} & \min_{1 \leq k \leq n} \left( (W^a W^b)_{ik} + W^c_{kj} \right) & \min_{1 \leq k \leq n} \left( W^a_{ik} + (W^b W^c)_{kj} \right) \\ = & \min_{1 \leq k \leq n} \left( \min_{1 \leq q \leq n} (W^a_{iq} + W^b_{qk}) + W^c_{kj} \right) & = \min_{1 \leq k \leq n} \left( W^a_{ik} + \min_{1 \leq q \leq n} (W^b_{kq} + W^c_{qj}) \right) \end{aligned}$$

$Q$  : Why do we care about this?

$A$  : Repeated squaring.

SSSP from  $s$

$l_v^{(m)}$  : the length of a shortest path  $s \rightsquigarrow v$  consisting of  $\leq m$  edges

$$l_v^{(m)} = \min_{1 \leq u \leq n} \left\{ l_u^{(m-1)} + w_{uv} \right\}, \quad m \geq 2$$

$$l_v^{(1)} = w_{sv}$$

$$L^{(n-1)} \triangleq \left( l_v^{(n-1)} \right)$$

## SSSP as a Product of Matrices and a Vector (Problem 25.1-5)

To express SSSP as a **product** of matrices and a vector.

$$L^{(1)} = \langle w_{sv} \rangle_{v \in V}$$

$$L^{(2)} = L^{(1)} \cdot W$$

$$\dots = \dots$$

$$L^{(n-1)} = \left( \left( \left( L^{(1)} \cdot W \right) \cdot W \right) \cdot \dots \cdot W \right)$$

## SSSP as a Product of Matrices and a Vector (Problem 25.1-5)

What is the **relationship** between it and the Bellman-Ford algorithm?

$$l_v^{(m)} = \min_{1 \leq u \leq n} \left\{ l_u^{(i-1)} + w_{uv} \right\}, \quad m \geq 2$$

$$d_v^{(i)} = \min_{u \rightarrow v} \left\{ d_u^{(i-1)} + w_{uv} \right\}, \quad i \geq 1$$

$$d_v^{(0)} = \begin{cases} 0 & v = s \\ \infty & v \neq s \end{cases}$$

---

---

```
1: procedure BELLMAN-FORD-DP( $G, w, s$ )
2:    $d[0, s] \leftarrow 0$ 
3:   for  $(v \neq s) \in V$  do
4:      $d[0, v] \leftarrow \infty$ 
5:   for  $i \leftarrow 1$  to  $|V| - 1$  do
6:     for  $v \in V$  do
7:        $d[i, v] = \infty$ 
8:       for  $(u, v) \in E$  do
9:         if  $d[i, v] > d[i - 1, u] + w(u, v)$  then           ▷ Simplify?
10:           $d[i, v] = d[i - 1, u] + w(u, v)$ 
```

---

---

```
1: procedure BELLMAN-FORD-DP-SIMPLIFIED( $G, w, s$ )
2:    $d[s] \leftarrow 0$ 
3:   for  $(v \neq s) \in V$  do
4:      $d[v] \leftarrow \infty$ 
5:   for  $i \leftarrow 1$  to  $|V| - 1$  do
6:     for  $v \in V$  do
7:       for  $(u, v) \in E$  do
8:         if  $d[v] > d[u] + w(u, v)$  then                                 $\triangleright$  Relax!
9:            $d[v] = d[u] + w(u, v)$ 
```

---

---

```
1: procedure BELLMAN-FORD-WITHOUT-NE( $G, w, s$ )
2:   INIT-SINGLE-SOURCE( $G, s$ )
3:   for  $i \leftarrow 1$  to  $|V| - 1$  do
4:     for  $(u, v) \in E$  do
5:       RELAX( $u, v, w$ )
```

---



Bellman-Ford:  $L \cdot W$

SLOW-ALL-PAIRS-SHORTEST-PATHS:  $W \cdot W$

SLOW-ALL-PAIRS-SHORTEST-PATHS is  $n$  instances of Bellman-Ford,  
one for each source.

## SSSP as a Product of Matrices and a Vector (Problem 25.1-5)

To express SSSP as a **product** of matrices and a vector.

$$L^{(n-1)} = \left( \left( \left( L^{(1)} \cdot W \right) \cdot W \right) \cdot \dots \cdot W \right)$$

*Q* : Associative? Repeated Squaring?

## Negative-weight Cycle Detection (Problem 25.1-9)

To detect negative-weight cycle (NC) using  
FASTER-ALL-PAIRS-SHORTEST-PATHS.

## Minimum-length Negative-weight Cycle (Problem 25.1-10)

To find the length of a minimum-length negative-weight cycle (NC).





Office 302

Mailbox: H016

hfwei@nju.edu.cn