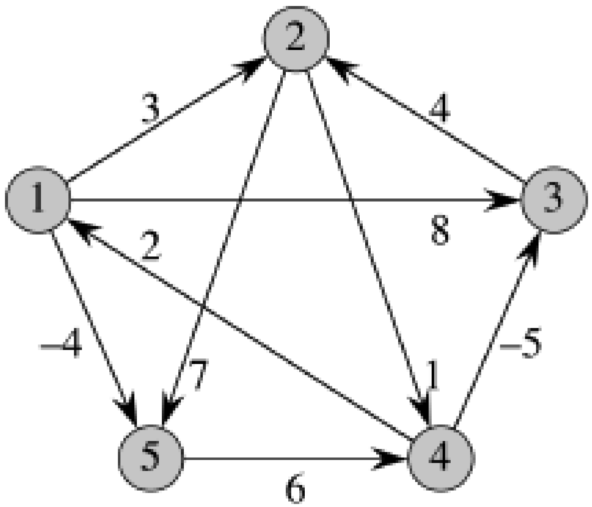
Spring 2012

ICS621 All Pairs Shortest Paths

Lipyeow Lim

Example



$$L^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$L^{(4)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

Dynamic Programming Solution 1

EXTEND(L, W, n)

let $L' = (l'_{ij})$ be a new $n \times n$ matrix
for $i = 1$ **to** n
  for $j = 1$ **to** n
    $l'_{ij} = \infty$
    for $k = 1$ **to** n
      $l'_{ij} = \min(l'_{ij}, l_{ik} + w_{kj})$
return L'

SLOW-APSP(W, n)

$L^{(1)} = W$
for $m = 2$ **to** $n - 1$
  let $L^{(m)}$ be a new $n \times n$ matrix
  $L^{(m)} = $ EXTEND$(L^{(m-1)}, W, n)$
return $L^{(n-1)}$

Time

- EXTEND: $\Theta(n^3)$.
- SLOW-APSP: $\Theta(n^4)$.

Dynamic Programming Solution 2

$$\begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \text{x} \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 3 & 8 & 2 & -4 \\ 3 & 0 & -4 & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & \infty & 1 & 6 & 0 \end{pmatrix}$$

EXTEND(L, W, n)

let $L' = (l'_{ij})$ be a new $n \times n$ matrix
for $i = 1$ **to** n
  for $j = 1$ **to** n
    $l'_{ij} = \infty$
    for $k = 1$ **to** n
      $l'_{ij} = \min(l'_{ij}, l_{ik} + w_{kj})$
return L'

L	$\rightarrow$	A
W	$\rightarrow$	B
L'	$\rightarrow$	C
min	$\rightarrow$	$+$
$+$	$\rightarrow$	$\cdot$
∞	$\rightarrow$	0

Faster Algo

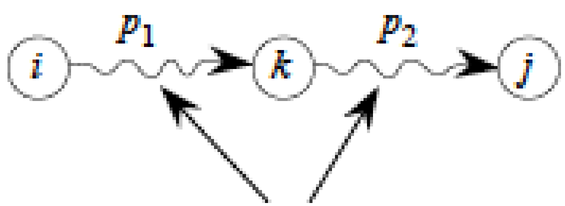
FASTER-APSP(W, n)

$L^{(1)} = W$
$m = 1$
while $m < n - 1$
  let $L^{(2m)}$ be a new $n \times n$ matrix
  $L^{(2m)} = \text{EXTEND}(L^{(m)}, L^{(m)}, n)$
  $m = 2m$
return $L^{(m)}$

Time

$\Theta(n^3 \lg n)$.

Floyd Warshall Algo



all intermediate vertices in $\{1, 2, ..., k-1\}$

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

FLOYD-WARSHALL(W, n)

$D^{(0)} = W$

for $k = 1$ **to** n

  let $D^{(k)} = \left(d_{ij}^{(k)}\right)$ be a new $n \times n$ matrix

  for $i = 1$ **to** n

    for $j = 1$ **to** n

      $d_{ij}^{(k)} = \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$

return $D^{(n)}$

Time

$\Theta(n^3)$.

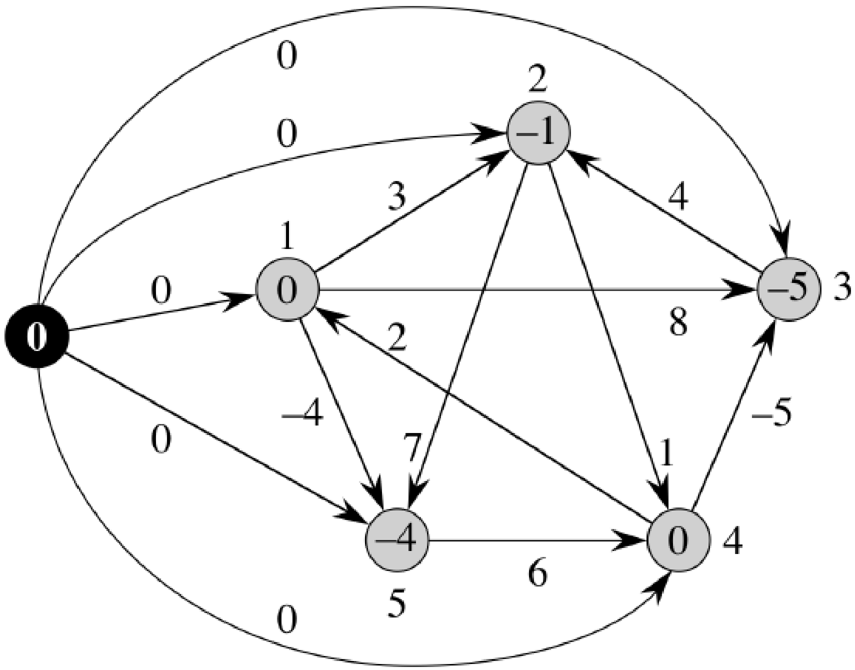
Johnson's Algo

Johnson's algorithm

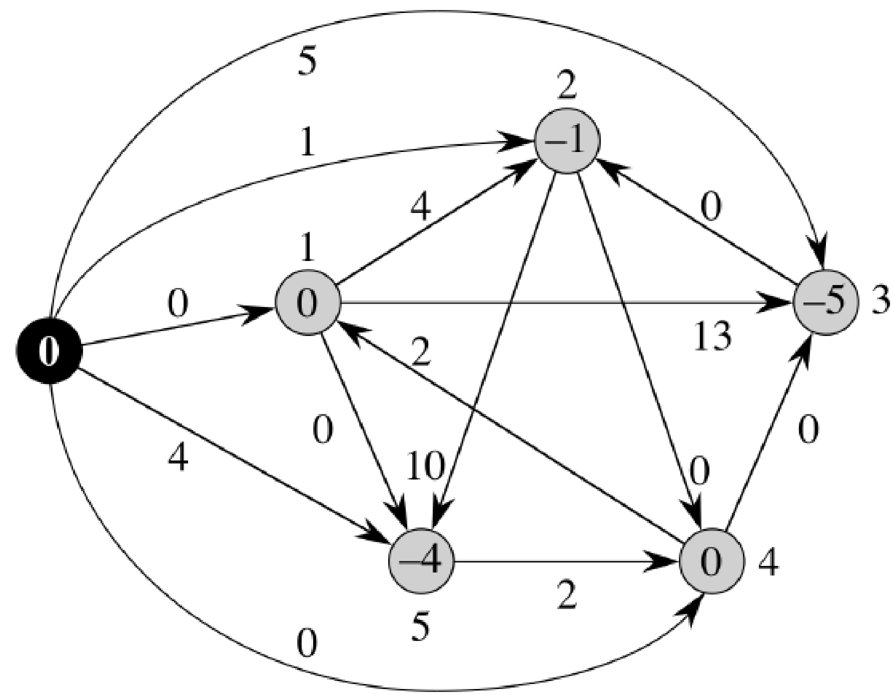
form G'
run BELLMAN-FORD on G' to compute $\delta(s, v)$ for all $v \in G'.V$
if BELLMAN-FORD returns FALSE
    G has a negative-weight cycle
else compute $\widehat{w}(u, v) = w(u, v) + \delta(s, u) - \delta(s, v)$ for all $(u, v) \in E$
    let $D = (d_{uv})$ be a new $n \times n$ matrix
    for each vertex $u \in G.V$
        run Dijkstra's algorithm from u using weight function $\widehat{w}$
            to compute $\widehat{\delta}(u, v)$ for all $v \in V$
        for each vertex $v \in G.V$
            // Compute entry d_{uv} in matrix D.
            $$d_{uv} = \underbrace{\widehat{\delta}(u, v) + \delta(s, v) - \delta(s, u)}_{\text{because if } p \text{ is a path } u \leadsto v,\text{ then } \widehat{w}(p) = w(p) + h(u) - h(v)}$$
    return D

$O(V^2 \lg V + VE)$.

Construction of G



(a)



(b)