Data Structures and Algorithms

Prof. Ganesh Ramakrishnan,
Prof. Ajit Diwan,
Prof. D.B. Phatak

Department of Computer Science and Engineering
IIT Bombay

Session: Shortest Path Algorithm
(All Pair Shortest Path)

All Pair Shortest Path Problem

1. Given: A directed weighted graph $G(V, E)$, for each edge $(v_1, v_2) \in E$ and an associated weight of an edge $w(v_1, v_2)$
2. Find: A shortest path from v_1 to v_2 for every pair of vertices v_1 and v_2 in V

Floyd-Warshall Algorithm: Setting up Notation

1. Given a directed weighted graph $G(V, E)$, the weight of each edge $(v_1, v_2) \in E$ can be defined using an an adjacency-matrix representation W
2. If W is an n x n matrix,

$$w_{v_i v_j} = \begin{cases} 0 & \text{if i} = \text{j}, \\ \text{weight of directed edge } (v_i,\ v_j) & \text{if i} \neq \text{j and (i, j)} \in \text{E}, \\ \infty & \text{if i} \neq \text{j OR (i, j)} \notin \text{E}. \end{cases}$$

3. Distance matrix $D^0 = W$
4. Predecessor matrix L can be defined as

$$L^0(i, j) = \begin{cases} \text{i} & \text{if (i, j)} \in \text{E} \\ \text{NULL} & \text{if i} = \text{j and (i, j)} \notin \text{E}. \end{cases}$$

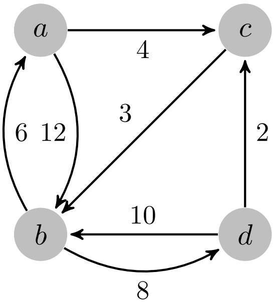
Floyd-Warshall Algorithm

Algorithm Floyd-WarshallAPSP(G, W)
$n = rows(W)$
$D^0 = W$
for $k \in n$ **do**
  let $D^k = d_{ij}^k$ be a new n x n matrix
  for $i \in n$ **do**
    for $j \in n$ **do**
      $d_{ij}^k = min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$
    end for
  end for
end for
return D^n

Figure: Floyd-Warshall Algorithm

Example

Graph

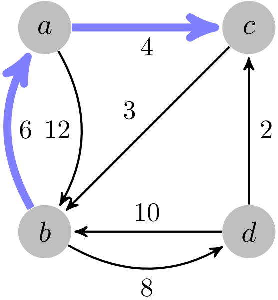


k=0

	a	b	c	d
a	0	12	4	∞
b	6	0	∞	8
c	∞	3	0	∞
d	∞	10	2	0

Example

Graph

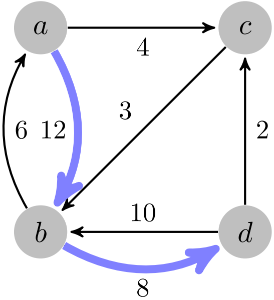


k = 1 (v_i =a)

	a	b	c	d
a	0	12	4	∞
b	6	0	**10**	8
c	∞	3	0	∞
d	∞	10	2	0

Example

Graph

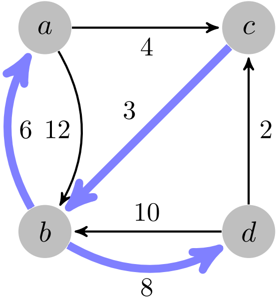


k $= 2$ (v_i =a)

	a	b	c	d
a	0	12	4	**20**
b	6	0	10	8
c	∞	3	0	∞
d	∞	10	2	0

Example

Graph

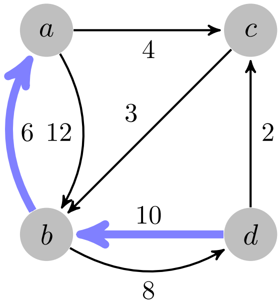


k $= 2$ (v_i =c)

	a	b	c	d
a	0	12	4	20
b	6	0	10	8
c	**9**	3	0	**11**
d	∞	10	2	0

Example

Graph

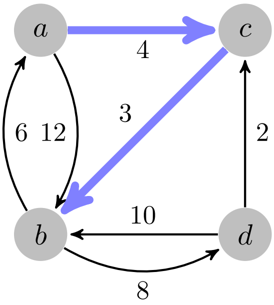


k = 2 (v_i =b)

	a	b	c	d
a	0	12	4	20
b	6	0	10	8
c	9	3	0	11
d	**16**	10	2	0

Example

Graph

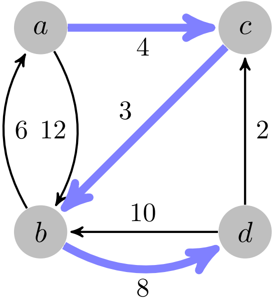


k = 3 (v_i =a, v_j =b)

	a	b	c	d
a	0	**7**	4	20
b	6	0	10	8
c	9	3	0	11
d	16	10	2	0

Example

Graph



k = 3 (v_i =a, v_j =d)

	a	b	c	d
a	0	7	4	**15**
b	6	0	10	8
c	9	3	0	11
d	16	10	2	0

Thank you