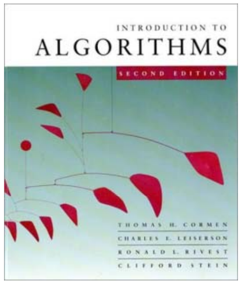


CMPS 2200 – Fall 2012

All Pairs Shortest Paths

Carola Wenk

Slides courtesy of Charles Leiserson
with changes by Carola Wenk

Shortest paths

Single-source shortest paths

- Nonnegative edge weights
  - Dijkstra's algorithm: $O(|E| \log |V|)$
- General: Bellman-Ford: $O(|V||E|)$
- DAG: One pass of Bellman-Ford: $O(|V| + |E|)$

Shortest paths

Single-source shortest paths

- Nonnegative edge weights
  - Dijkstra's algorithm: $O(|E| \log |V|)$
- General: Bellman-Ford: $O(|V||E|)$
- DAG: One pass of Bellman-Ford: $O(|V| + |E|)$

All-pairs shortest paths

- Nonnegative edge weights
  - Dijkstra's algorithm $|V|$ times: $O(|V||E| \log |V|)$
- General
  - Bellman-Ford $|V|$ times: $O(|V|^2|E|)$
  - Floyd-Warshall: $O(|V|^3)$

All-pairs shortest paths

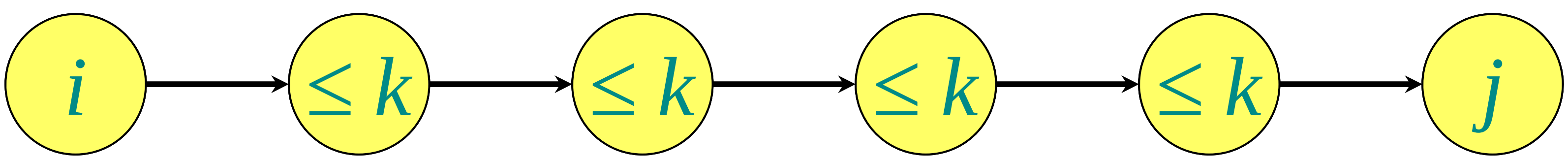
Input: Digraph $G = (V, E)$, where $|V| = n$, with edge-weight function $w : E \rightarrow \mathbb{R}$.
Output: $n \times n$ matrix of shortest-path lengths $\delta(i, j)$ for all $i, j \in V$.

Algorithm #1:
- Run Bellman-Ford once from each vertex.
- Time $= O(|V|^2|E|)$.
- But: Dense graph $\Rightarrow$ $O(|V|^4)$ time.

Floyd-Warshall algorithm

- Dynamic programming algorithm.
- Assume $V=\{1, 2, \ldots, n\}$, and assume G is given in an **adjacency matrix** $A=(a_{ij})_{1\le i,j\le n}$ where a_{ij} is the weight of the edge from i to j.

Define $c_{ij}^{(k)}$ = weight of a shortest path from i to j with intermediate vertices belonging to the set $\{1, 2, \ldots, k\}$.



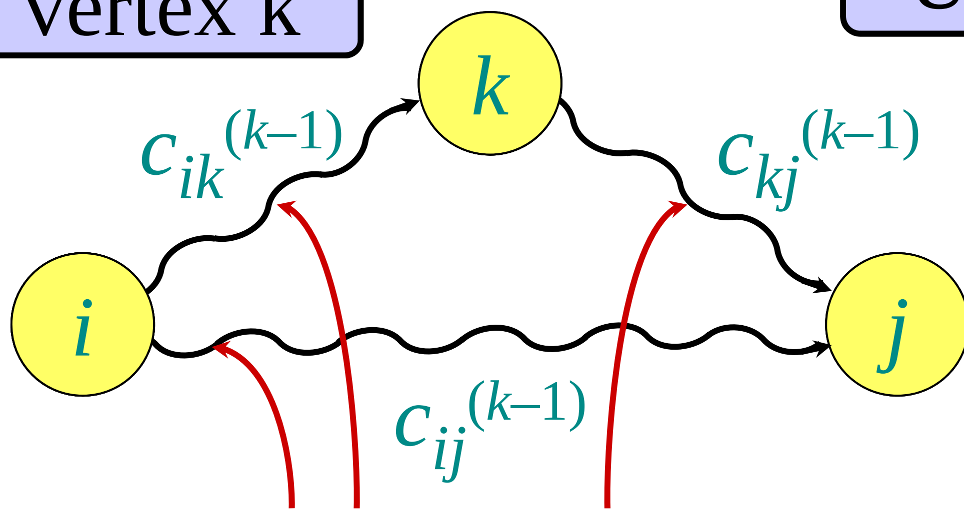
Thus, $\delta(i, j) = c_{ij}^{(n)}$. Also, $c_{ij}^{(0)} = a_{ij}$.

Floyd-Warshall recurrence

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

Do not use vertex k

Use vertex k



intermediate vertices in $\{1, 2, \ldots, k\text{-}1\}$

Pseudocode for Floyd-Warshall

for $k \leftarrow 1$ **to** n **do**
  for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow 1$ **to** n **do**
      if $c_{ij}^{(k-1)} > c_{ik}^{(k-1)} + c_{kj}^{(k-1)}$ **then**
        $c_{ij}^{(k)} \leftarrow c_{ik}^{(k-1)} + c_{kj}^{(k-1)}$
      else
        $c_{ij}^{(k)} \leftarrow c_{ij}^{(k-1)}$

} ***relaxation***

- Runs in $\Theta(n^3)$ time and space
- Simple to code.
- Efficient in practice.

Shortest paths

Single-source shortest paths

- Nonnegative edge weights
  - Dijkstra's algorithm: $O(|E| \log |V|)$
- General: Bellman-Ford: $O(|V||E|)$
- DAG: One pass of Bellman-Ford: $O(|V| + |E|)$

} adj. list

All-pairs shortest paths

- Nonnegative edge weights — adj. list
  - Dijkstra's algorithm $|V|$ times: $O(|V||E| \log |V|)$
- General
  - Bellman-Ford $|V|$ times: $O(|V|^2|E|)$ — adj. list
  - Floyd-Warshall: $O(|V|^3)$ — adj. matrix