All Pairs Shortest Paths

CSCI 3100

Shortest Path between all pairs

Brute force solution:

- Calculate all simple paths between every pair of vertices
  - Number of pairs of vertices: $\binom{|V|}{2} = \frac{1}{2}(|V|^2 - |V|) = O(|V|^2)$
  - Number of simple paths between two vertices: worst case $O(2^{|V|})$
- Select the shortest path
- Worst case complexity: $O(|V|^2 2^{|V|})$

Repeated Bellman-Ford solution

- Run Bellman-Ford with a new starting vertex each time, total of |V| runs
- Complexity: $O(|V| * |V| * |E|)$
- Worst case: $O(|V|^4), if\ |E| \approx |V|^2$

Dynamic Programming Solutions

Repeated squaring: $O(|V|^3 \lg(|V|)$ - Ch 25.1

Floyd-Warshall algorithm: $O(|V|^3)$ – Ch 25.2

Graph is represented as an adjacency matrix W

W[x, y] is the weight of edge (x, y)

Produce a matrix with shortest path distances between any pairs of vertices

Optimal Substructure: Floyd-Warshall

Let (v_1, v_2, …, v_m) be a shortest path from v_1 to v_m

Consider an *intermediate* vertex of this path: any vertex in the path other than v_1 and v_m

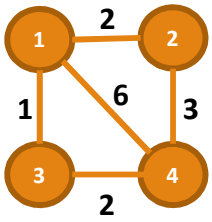
Let vertices of G be V = {1, 2, …, N}

Consider a subset of vertices {1, 2, …, k} for some k

For any pair of vertices *x, y*, consider all paths where intermediate vertices are in the set {1, 2, …, k}

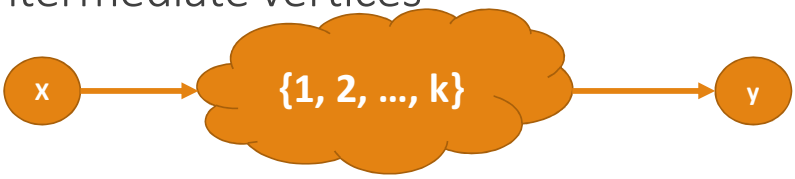
Let p be the shortest of such paths

Example



- Let x = 1, y = 4
- Let k = 0: intermediate set is empty
  - 1->4: weight 6
- Let k = 1: intermediate set {1}
  - 1->4: weight 6
- Let k = 2: intermediate set {1, 2}
  - 1->4: total weight 6
  - 1->2->4: total weight 5
- Let k = 3: intermediate set {1, 2, 3}
  - 1->4: total weight 6
  - 1->2->4: total weight 5
  - 1->3->4: total weight 3

What can we observe about a shortest path p from *x to y* built from intermediate vertices {1, 2, …, k} only?



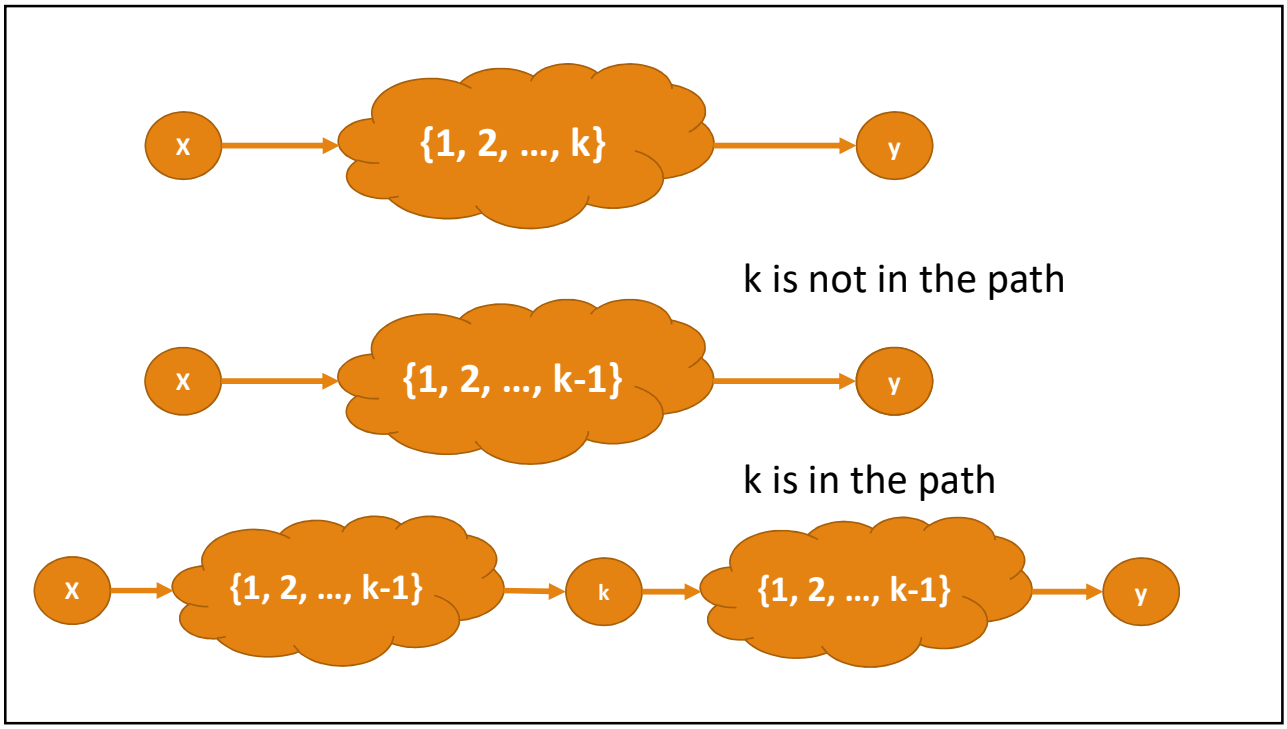
A. If k is not in path p, then all vertices of path p are from the set {1, 2, …, k-1}

B. If k is in path p, then there is a shortest path from *x to k* using only vertices from the set {1, 2, …, k-1}

C. If k is in path p, then there is a shortest path from *k to y* using only vertices from the set {1, 2, …, k-1}

D. All of the above

E. None of the above



Optimal substructure

Let $D^k[x, y]$ be the weight of the shortest path from x to y using vertices {1, 2, …, k}

When k = 0, no intermediate vertices are used, so $D^0[x, y] = W[x, y]$

When k > 0

- If k is in the shortest path from x to y, then $Dk[x, y] = D^{k-1}[x, k] + D^{k-1}[k, y]$
- If k is not in the shortest path from x to y, then $D^k[x, y] = D^{k-1}[x, y]$

Recursive definition:

$$D^k[x,y] = \begin{cases} W[x,y] \;\; if \; k = 0 \\ \min(D^{k-1}[x,y], \;\; D^{k-1}[x,k] + D^{k-1}[k,y]) \; if \; k > 0 \end{cases}$$

Compute D^k using "bottom up" approach

Compute $D^0 = W$

Compute D^1 using D^0

…

Compute D^N using D^{N-1}

Return D^N

```
FLOYD-WARSHALL (W)
1 N = W.rows
2 D^0 = W
3 for k = 1 to N
4 D^k  is an N x N matrix
5 for x = 1 to N
6    for y = 1 to N
7      D^k[x,y] = min(D^k-1[x,y], D^k-1[x,k]+D^k-1[k,y])
8 Return D^N
```

Complexity:

Reconstructing shortest paths

Modify the algorithm to compute predecessor matrix P^N

$P^N[x, y] = k$ means that the last edge in the shortest path from x to y is (k, y)

Floyd-Warshall algorithm has the following line for computing the shortest path distance between vertices x and y:

$D^k[x,y] = \min(D^{k-1}[x,y], D^{k-1}[x,k]+D^{k-1}[k,y])$

We want to compute predecessor matrix P^k. When k = 0, then $P^k[x, y]=x$, if (x, y) is an edge. Otherwise $P^k[x, y]$ = NULL. How can we calculate $P^k[x, y]$ when k > 0?

A. $P^{k-1}[x,y]$ if $D^{k-1}[x,y] < D^{k-1}[x,k]+D^{k-1}[k,y])$

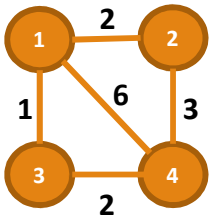
B. $P^{k-1}[k,y]$ if $D^{k-1}[x,k]+D^{k-1}[k,y] < D^{k-1}[x,y]$

C. $P^{k-1}[x,k]$ if $D^{k-1}[x,k]+D^{k-1}[k,y] < D^{k-1}[x,y]$

D. A and B

E. A and C

Example



K = 0, D^0 = $\begin{matrix} 0 & 2 & 1 & 6 \\ 2 & 0 & \infty & 3 \\ 1 & \infty & 0 & 2 \\ 6 & 3 & 2 & 0 \end{matrix}$ P^0 = $\begin{matrix} N & 1 & 1 & 1 \\ 2 & N & N & 2 \\ 3 & N & N & 3 \\ 4 & 4 & 4 & N \end{matrix}$

K = 1, D^1 = $\begin{matrix} 0 & 2 & 1 & 6 \\ 2 & 0 & \mathbf{3} & 3 \\ 1 & \mathbf{3} & 0 & 2 \\ 6 & 3 & 2 & 0 \end{matrix}$ P^1 = $\begin{matrix} N & 1 & 1 & 1 \\ 2 & N & \mathbf{1} & 2 \\ 3 & \mathbf{1} & N & 3 \\ 4 & 4 & 4 & N \end{matrix}$

K = 2, D^2 = $\begin{matrix} 0 & 2 & 1 & \mathbf{5} \\ 2 & 0 & 3 & 3 \\ 1 & 3 & 0 & 2 \\ \mathbf{5} & 3 & 2 & 0 \end{matrix}$ P^2 = $\begin{matrix} N & 1 & 1 & \mathbf{2} \\ 2 & N & 1 & 2 \\ 3 & 1 & N & 3 \\ \mathbf{2} & 4 & 4 & N \end{matrix}$

K = 3, D^3 = $\begin{matrix} 0 & 2 & 1 & \mathbf{3} \\ 2 & 0 & 3 & 3 \\ 1 & 3 & 0 & 2 \\ \mathbf{3} & 3 & 2 & 0 \end{matrix}$ P^3 = $\begin{matrix} N & 1 & 1 & \mathbf{3} \\ 2 & N & 1 & 2 \\ 3 & 1 & N & 3 \\ \mathbf{3} & 4 & 4 & N \end{matrix}$

K = 4, D^4 = $\begin{matrix} 0 & 2 & 1 & \mathbf{3} \\ 2 & 0 & 3 & 3 \\ 1 & 3 & 0 & 2 \\ \mathbf{3} & 3 & 2 & 0 \end{matrix}$ P^4 = $\begin{matrix} N & 1 & 1 & \mathbf{3} \\ 2 & N & 1 & 2 \\ 3 & 1 & N & 3 \\ \mathbf{3} & 4 & 4 & N \end{matrix}$