Advanced Algorithms & Data Structures

Department of CSE

ADVANCED ALGORITHMS AND DATA STRUCTURES
23CS03HF

Topic:

All Shortest Path Problem

Session - 24

KL ACCREDITED BY NAAC WITH A++ GRADE
nirf 2022 RANKED 27 AMONG ALL UNIVERSITIES
CATEGORY 1 UNIVERSITY BY MHRD, Govt. of India
43 YEARS OF EDUCATIONAL LEADERSHIP

AIM OF THE SESSION

To familiarize students with the concept of All pairs Shortest Path Problem

INSTRUCTIONAL OBJECTIVES

This Session is designed to:

1.Demonstrate :- All Pairs Shortest Path Problem.

2.Describe :- Solve All pairs shortest path problem using Floyd-Warshall Algorithm .

LEARNING OUTCOMES

At the end of this session, you should be able to:

1. Define :- All Pairs Shortest path Problem.
2. Describe :- solve All Pairs shortest path problem using Floyd-Warshall Algorithm
3. Summarize:- Finding the shortest paths from all source vertices to all other vertices in a weighted graph.

- ***The problem:*** **find the shortest path between every pair of vertices of a graph**
- ***The graph*****: may contain negative edges but no negative cycles**
- ***A representation*****: a weight matrix where**
  W(i,j)=0 if i=j.
  W(i,j)=∞ if there is no edge between i and j.
  W(i,j)="weight of edge"

All Pairs Shortest Path

- ***The problem:*** Find the shortest path between every pair of vertices of a graph
- *The graph*: may contain negative edges but no negative cycles
- *A representation*: a weight matrix where
  W(i,j)=0 if i=j.
  W(i,j)=∞ if there is no edge between i and j.
  W(i,j)="weight of edge"

KL ACCREDITED BY NAAC WITH A++ GRADE
nirf 2022 RANKED 27 AMONG ALL UNIVERSITIES
CATEGORY 1 UNIVERSITY BY MHRD, Govt. of India
43 YEARS OF EDUCATIONAL LEADERSHIP

Floyd-Warshall Algorithm

- Let $D^{(k)}[i,j]$=weight of a shortest path from v_i to v_j using only vertices from $\{v_1,v_2,\ldots,v_k\}$ as intermediate vertices in the path
  - $D^{(0)}=W$
  - $D^{(n)}=D$ which is the goal matrix
- How do we compute $D^{(k)}$ from $D^{(k-1)}$?

KL ACCREDITED BY NAAC WITH A++ GRADE
nirf 2022 RANKED 27 AMONG ALL UNIVERSITIES
CATEGORY 1 UNIVERSITY BY MHRD, Govt. of India
43 YEARS OF EDUCATIONAL LEADERSHIP

Conditions

Case 1: A shortest path from v_i to v_j restricted to using only vertices from $\{v_1,v_2,\ldots,v_k\}$ as intermediate vertices does not use v_k.

Then $D^{(k)}[i,j]= D^{(k-1)}[i,j]$.

Case 2: A shortest path from v_i to v_j restricted to using only vertices from $\{v_1,v_2,\ldots,v_k\}$ as intermediate vertices does use v_k.

Then $D^{(k)}[i,j]= D^{(k-1)}[i,k]+ D^{(k-1)}[k,j]$.

The recursive definition

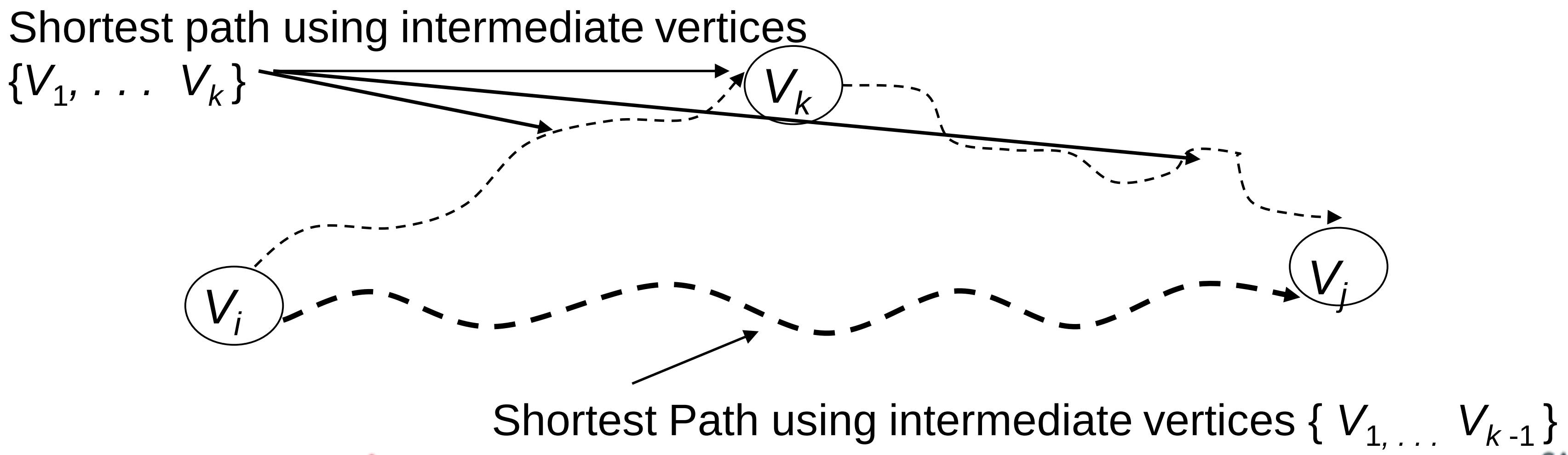
- Since

  $D^{(k)}[i,j]= D^{(k-1)}[i,j]$ or

  $D^{(k)}[i,j]= D^{(k-1)}[i,k]+ D^{(k-1)}[k,j]$.

- We conclude:

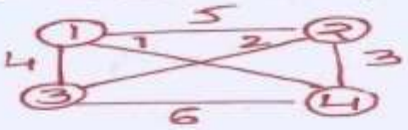
  $D^{(k)}[i,j]= \min\{ D^{(k-1)}[i,j], D^{(k-1)}[i,k]+ D^{(k-1)}[k,j] \}$.



KL ACCREDITED BY NAAC WITH A++ GRADE
nirf 2022 RANKED 27 AMONG ALL UNIVERSITIES
CATEGORY 1 UNIVERSITY BY MHRD, Govt. of India
43 YEARS OF EDUCATIONAL LEADERSHIP

Implementation

Eg. Find the shortest distance from each vertex to other vertices in the below graph.



The All Pairs shortest-path problem is to determine a matrix A such that $A(i, j)$ is the length of a shortest path i, j.

From floyd's alg. we have the recurrence relation :

$$A(i,j) = \min\left\{ \min_{1 \le k \le n} \{ A^{k-1}(i,k) + A^{k-1}(k,j) \}, cost(i,j) \right\}$$

$$\Rightarrow A^{0}(i,j) = cost(i,j), \quad 1 \le i \le n, \quad 1 \le j \le n$$

$$\boxed{A^{k}(i,j) = \min\{ A^{k-1}(i,j), A^{k-1}(i,k) + A^{k-1}(k,j) \}, \quad k \ge 1}$$

The cost matrix for given graph is given by

$$A^{0} = cost(i,j) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 5 & 4 & 1 \\ 5 & 0 & 2 & 3 \\ 4 & 2 & 0 & 6 \\ 1 & 3 & 6 & 0 \end{bmatrix} \end{array}$$

where $cost(i,j) = \begin{cases} 0 & \text{if } i = j \\ \text{edge } cost(i,j) & \text{if } i \ne j \ \& \ (i,j) \in E \\ \infty & \text{if } i \ne j \ \& \ (i,j) \notin E \end{cases}$

and E is set of edges in graph 'G'.

When computing $A^{(1)}$,
- 1^{st} row & 1^{st} column can be copied from $A^{(0)}$ to $A^{(1)}$ as they remain constant.
- all diagonal elements will be 0's always.

$k \leftarrow 1$, $i \swarrow$, $j \swarrow$

$a^1_{23} = \min\{a^0_{23}, (a^0_{21} + a^0_{13})\}$
$= \min\{2, (5+4)\} = 2$

$a^1_{24} = \min\{a^0_{24}, (a^0_{21} + a^0_{14})\} = \min\{3, (5+1)\} = 3$

$a^1_{32} = \min\{a^0_{32}, (a^0_{31} + a^0_{12})\} = \min\{2, (4+5)\} = 2$

$a^1_{34} = \min\{a^0_{34}, (a^0_{31} + a^0_{14})\} = \min\{6, (4+1)\} = 5$

$a^1_{42} = \min\{a^0_{42}, (a^0_{41} + a^0_{12})\} = \min\{3, (1+5)\} = 3$

$a^1_{43} = \min\{a^0_{43}, (a^0_{41} + a^0_{13})\} = \min\{6, (1+4)\} = 5$

$$\therefore A^1 = \begin{bmatrix} 0 & 5 & 4 & 1 \\ 5 & 0 & 2 & 3 \\ 4 & 2 & 0 & \underline{5} \\ 1 & 3 & \underline{5} & 0 \end{bmatrix}$$

$$\text{lly } A^2 = \begin{bmatrix} 0 & 5 & 4 & 1 \\ 5 & 0 & 2 & 3 \\ 4 & 2 & 0 & 5 \\ 1 & 3 & 5 & 0 \end{bmatrix}$$

$$\text{lly } A^3 = \begin{bmatrix} 0 & 5 & 4 & 1 \\ 5 & 0 & 2 & 3 \\ 4 & 2 & 0 & 5 \\ 1 & 3 & 5 & 0 \end{bmatrix}$$

$$A^4 = \begin{bmatrix} 0 & \underline{4} & 4 & 1 \\ \underline{4} & 0 & 2 & 3 \\ 4 & 2 & 0 & 5 \\ 1 & 3 & 5 & 0 \end{bmatrix}$$

SUMMARY

- **Objective**: Find the shortest paths from all source vertices to all other vertices in a weighted graph.

- **Approach**:
  - Initialize the solution matrix same as the input graph matrix as a first step.
  - Then update the solution matrix by considering all vertices as an intermediate vertex.
  - The idea is to pick all vertices one by one and updates all shortest paths which include the picked vertex as an intermediate vertex in the shortest path.
  - When we pick vertex number **k** as an intermediate vertex, we already have considered vertices **{0, 1, 2, .. k-1}** as intermediate vertices.

- **Algorithmic Solutions**: Algorithms like Floyd Warshall algorithm used to solve this problem efficiently.

SELF-ASSESSMENT QUESTIONS

The Floyd-Warshall algorithm is based on which programming paradigm?

(a) Divide and conquer
(b) Greedy method
(c) Dynamic programming
(d) Backtracking

What is the time complexity of the Floyd-Warshall algorithm for a graph with n vertices?

(a) $O(n^3)$
(b) $O(n^2)$
(c) $O(n^4)$
(d) $O(n\log n)$

TERMINAL QUESTIONS

1. Given the adjacency matrix of a graph:

$$\begin{bmatrix} 0 & 3 & \infty & 7 \\ 8 & 0 & 2 & \infty \\ 5 & \infty & 0 & 1 \\ 2 & \infty & \infty & 0 \end{bmatrix}$$

Use the Floyd-Warshall algorithm to compute the shortest path matrix. Show all intermediate steps.

2. Describe the differences between the Floyd-Warshall algorithm and Dijkstra's algorithm for solving shortest path problems.

KL ACCREDITED BY NAAC WITH A++ GRADE
nirf 2022 NATIONAL INSTITUTIONAL RANKING FRAMEWORK RANKED 27 AMONG ALL UNIVERSITIES
CATEGORY 1 UNIVERSITY BY MHRD, Govt. of India
43 YEARS OF EDUCATIONAL LEADERSHIP

REFERENCES FOR FURTHER LEARNING OF THE SESSION

Reference Books :

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein., 3rd, 2009, The MIT Press.
2 Algorithm Design Manual, Steven S. Skiena., 2nd, 2008, Springer.
3 Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, and Michael H. Goldwasser., 2nd, 2013, Wiley.
4 The Art of Computer Programming, Donald E. Knuth, 3rd, 1997, Addison-Wesley Professiona.

MOOCS :

1. https://www.coursera.org/specializations/algorithms?=
2.https://www.coursera.org/learn/dynamic-programming-greedy-algorithms#modules

THANK YOU