

Floyd's Algorithm

- Floyd's algorithm gives the solution to all pairs shortest path problem.
- Computes the distance matrix of a weighted graph through series of matrices.

Pseudocode

- **Input:** The weight matrix of a graph with no negative-length cycle
 - **Output:** The distance matrix of the shortest paths' lengths
1. for $k \leftarrow 1$ to n do
 2. for $i \leftarrow 1$ to n do
 3. for $j \leftarrow 1$ to n do
 4. $D[i, j] \leftarrow \min \{D[i, j], D[i, k] + D[k, j]\}$
 5. return D

Time Complexity of Floyd's Algorithm

- The basic operation is in the inner most for loop. It executes for every iteration.
- There are three for loops in this algorithm.
- Each for loop executes n times for every iteration of its outer for loop.
- Number of times the basic operation executes is given by the expression

$$\begin{aligned} C(n) &= n \times n \times n \\ &= n^3 \end{aligned}$$

- **Time complexity of Floyd's algorithm is $\Theta(n^3)$.**
- It is same for best case, worst case and average case scenarios.