

Floyd-Warshall's Algorithm (All Points Shortest Path)¹

Input: A weighted graph $G = (V, E)$ with weight function $f : V \times V \rightarrow \mathcal{R}^+ \cup \{+\infty\}$.

Let n denote the number of vertices in V . Number the vertices $V = \{v_1, v_2, v_3, \dots, v_n\}$.

Output: An $n \times n$ matrix C such that $C_{i,j}$ is the cost of the shortest path from v_i to v_j .

Key Idea: Define a matrix $C_{i,j}^{(k)}$ as the cost of the shortest (restricted) path from v_i to v_j that goes through intermediate vertices numbered no higher than k .

Method:

```

for ( i = 1 ; i ≤ n ; i++ )
    Ci,i(0) = 0
// end for i

for ( i = 1 ; i ≤ n ; i++ )
    for ( j = 1 ; j ≤ n ; j++ )
        if ( i ≠ j )
            Ci,j(0) = f(vi, vj)
        // end if
    end for j
end for i

// End of initialization phase. Begin main loop.
for ( k = 1 ; k ≤ n ; k++ )
    for ( i = 1 ; i ≤ n ; i++ )
        for ( j = 1 ; j ≤ n ; j++ )
            Ci,j(k) = min ( Ci,j(k-1) , ( Ci,k(k-1) + Ck,j(k-1) ) )
        // end for j
    // end for i
// end for k
Let C = Ci,j(n)
Return C.

```

¹A version of this algorithm could be set to rock and roll music. That would be “Pink Floyd-Warshall”.