1.Warshalls
2.Floyds
3.Compute time complexity for both

```c
#include <stdio.h>
#define INF 99999
#define V 4

void warshall(int graph[V][V]) {
    int i, j, k;

    for (k = 0; k < V; k++) {
        for (i = 0; i < V; i++) {
            for (j = 0; j < V; j++) {
                graph[i][j] = graph[i][j] || (graph[i][k] && graph[k][j]);
            }
        }
    }
}

void floydWarshall(int graph[V][V]) {
    int i, j, k;

    for (k = 0; k < V; k++) {
        for (i = 0; i < V; i++) {
            for (j = 0; j < V; j++) {
                if (graph[i][j] > graph[i][k] + graph[k][j]) {
                    graph[i][j] = graph[i][k] + graph[k][j];
                }
            }
        }
    }
}

void printGraph(int graph[V][V]) {
    for (int i = 0; i < V; i++) {
        for (int j = 0; j < V; j++) {
            if (graph[i][j] == INF)
                printf("INF ");
            else
```

```
                printf("%d ", graph[i][j]);
            }
            printf("\n");
        }
}

int main() {

    int graph1[V][V] = {
        {1, 1, 0, 0},
        {0, 1, 1, 0},
        {0, 0, 1, 1},
        {0, 0, 0, 1}
    };

    int graph2[V][V] = {
        {0, 3, INF, INF},
        {2, 0, INF, INF},
        {INF, 7, 0, 1},
        {6, INF, INF, 0}
    };

    printf("Original Graph for Warshall's Algorithm (Adjacency
Matrix):\n");
    printGraph(graph1);

    warshall(graph1);

    printf("\nTransitive Closure of the Graph (Warshall's Algorithm):\n");
    printGraph(graph1);

    printf("\nOriginal Graph for Floyd-Warshall Algorithm (Adjacency
Matrix):\n");
    printGraph(graph2);

    floydWarshall(graph2);

    printf("\nShortest Paths between all pairs of vertices (Floyd-Warshall
Algorithm):\n");
    printGraph(graph2);
```

```
    return 0;
}
```

Output

```
Original Graph for Warshall's Algorithm (Adjacency Matrix):
1 1 0 0
0 1 1 0
0 0 1 1
0 0 0 1

Transitive Closure of the Graph (Warshall's Algorithm):
1 1 1 1
0 1 1 1
0 0 1 1
0 0 0 1

Original Graph for Floyd-Warshall Algorithm (Adjacency Matrix):
0 3 INF INF
2 0 INF INF
INF 7 0 1
6 INF INF 0

Shortest Paths between all pairs of vertices (Floyd-Warshall Algorithm):
0 3 INF INF
2 0 INF INF
7 7 0 1
6 9 INF 0
PS C:\Users\STUDENT>
```