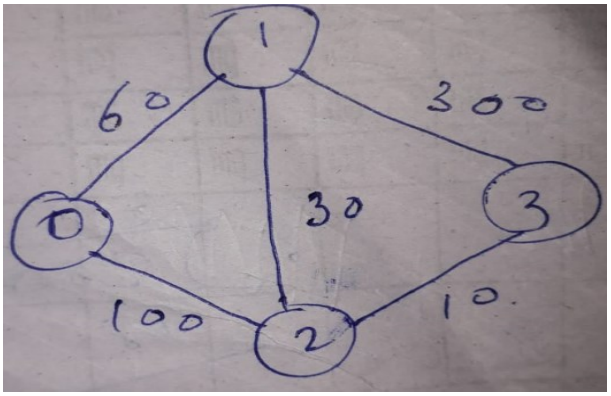
NAME:	SANIYA BANGARE
UID:	2021300009
BATCH:	A
SUBJECT:	DAA
EXP NO:	5
AIM:	Experiment based on greedy approach (Dijkstra's Algorithm)
THEORY:	Dijkstra's algorithm allows us to find the shortest pathbetween any two vertices of a graph. It differs from the minimum spanning tree because the shortest distance between two vertices might not include allthe vertices of the graph. Dijkstra's Algorithm works on the basis that any subpath B -> D of the shortest path A -> D between vertices A and D is also the shortest path between vertices B and D. Djikstra used this property in the opposite direction i.e we overestimate the distance of each vertex from the starting vertex. Then we visit each node and its neighbors to find the shortest subpath to those neighbors. The algorithm uses a greedy approach in the sense that we find the next best solution hoping that the end result is the best solution for the whole problem. Example: 

PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
#include<limits.h>
#include<stdbool.h>

int minDistance(int dist[], bool minSet[], int V)
{
   int min = INT_MAX, min_index;

   for (int v = 0; v < V; v++)
     if (minSet[v] == false && dist[v] <= min)
         min = dist[v], min_index = v;

   return min_index;
}

void printSolution(int dist[], int V)
{
   printf("Vertex\t\tDistance from Source \n");
    for (int i = 0; i < V; i++)
        printf("%d \t\t\t %d\n", i, dist[i]);
}

void dijkstra( int v, int graph[v][v], int source){
    int dist[v];
    bool minSet[v];
    for(int i=0;i<v;i++){
        dist[i]=INT_MAX;
        minSet[i]=false;
    }
    dist[source]=0;
    for(int count=0;count<v-1;count++){
        int u = minDistance(dist,minSet,v);
        minSet[u]=true;
        for(int i=0;i<v;i++){
            if(!minSet[i] && graph[u][i] && dist[u]!=INT_MAX
&& dist[u]+graph[u][i]<dist[i]){
                dist[i]=dist[u]+graph[u][i];
            }
        }
    }
    printSolution(dist,v);
}

int main(){
    int v;
    printf("Enter the number of vertices: ");
    scanf("%d",&v);
    int graph[v][v];
```

```
    printf("Enter the adjacency matrix: ");
    for(int i=0;i<v;i++){
        for(int j=0;j<v;j++){
            scanf("%d",&graph[i][j]);
        }
    }
    dijkstra(v,graph,0);
    return 0;

}
```

RESULT:

```
e:\c tutorial\output>.\"dijkstra.exe"
Enter the number of vertices: 4
Enter the adjacency matrix: 0 60 100 0
60 0 30 300
100 30 0 10
0 300 10 0
Vertex          Distance from Source
0                       0
1                       60
2                       90
3                       100
```

CONCLUSION:

From this experiment, I learnt about Dijkstra algorithm and understood how greedy approach is used. And how to find the shortest possible path between the source and other nodes.