Analysis & Design of Algorithms

Semester III 2018-19

Lab - 8

Topics: To find the Shortest Path (SP) using Dijkstra's algorithm

---

INTRODUCTION

In this lab we would be finding Shortest Path (SP) using Dijkstra's algorithm. Graph is represented using adjacency list or adjacency matrix. The implementation can be also done using adjacency list/ matrix. The code for constructing an graph is using adjacency matrix is given below.

Graph using adjacency list

```
#include <stdio.h>
#include <limits.h>
// Number of vertices in the graph
#define V 9
// A utility function to find the vertex with minimum distance value, from
// the set of vertices not yet included in shortest path tree
int minDistance(int dist[], bool sptSet[]) {
        // Initialize min value
        int min = INT_MAX, min_index;
        for (int v = 0; v < V; v++)
                if (sptSet[v] == false && dist[v] <= min)
                min = dist[v], min_index = v;
                return min_index;
}
// A utility function to print the constructed distance array
int printSolution(int dist[], int n) {
        printf("Vertex␣Distance␣from␣Source\n");
        for (int i = 0; i < V; i++)
        printf("%d␣tt␣%d\n", i, dist[i]);
}
```

```
// Function that implements Dijkstra's algorithm
void dijkstra(int graph[V][V], int src) {
// write the code for dijkstra
}

// Calling main function
int main() {
/* Let us create the example graph discussed below */
        int graph[V][V] = {{0, 4, 0, 0, 0, 0, 0, 8, 0},
                           {4, 0, 8, 0, 0, 0, 0, 11, 0},
                           {0, 8, 0, 7, 0, 4, 0, 0, 2},
                           {0, 0, 7, 0, 9, 14, 0, 0, 0},
                           {0, 0, 0, 9, 0, 10, 0, 0, 0},
                           {0, 0, 4, 14, 10, 0, 2, 0, 0},
                           {0, 0, 0, 0, 0, 2, 0, 1, 6},
                           {8, 11, 0, 0, 0, 0, 1, 0, 7},
                           {0, 0, 2, 0, 0, 0, 6, 7, 0} };
        dijkstra(graph, 0);
        return 0;
}
```

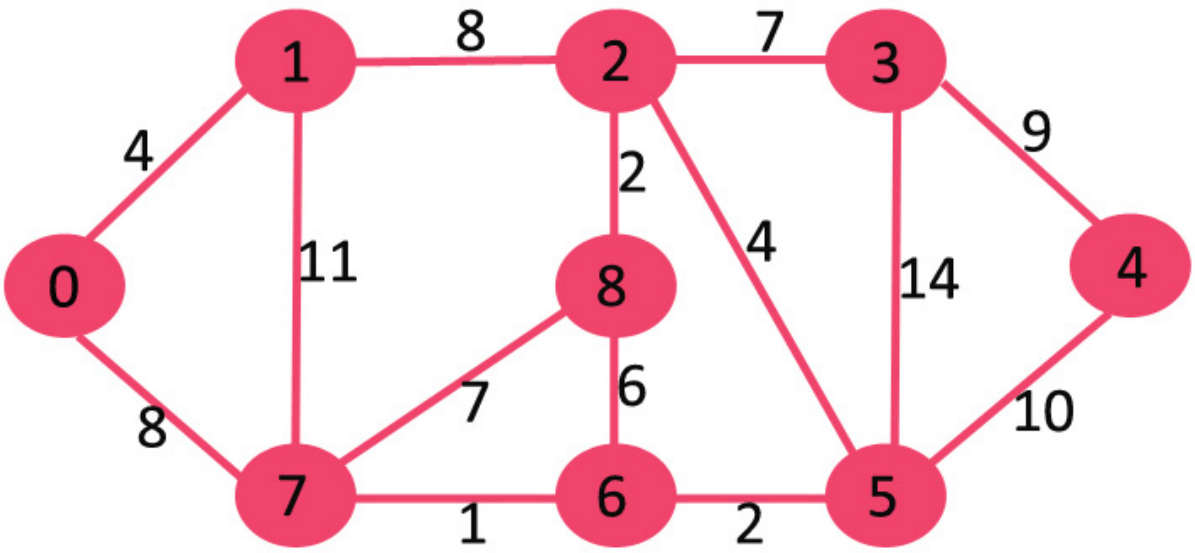

Figure 1: Given Graph

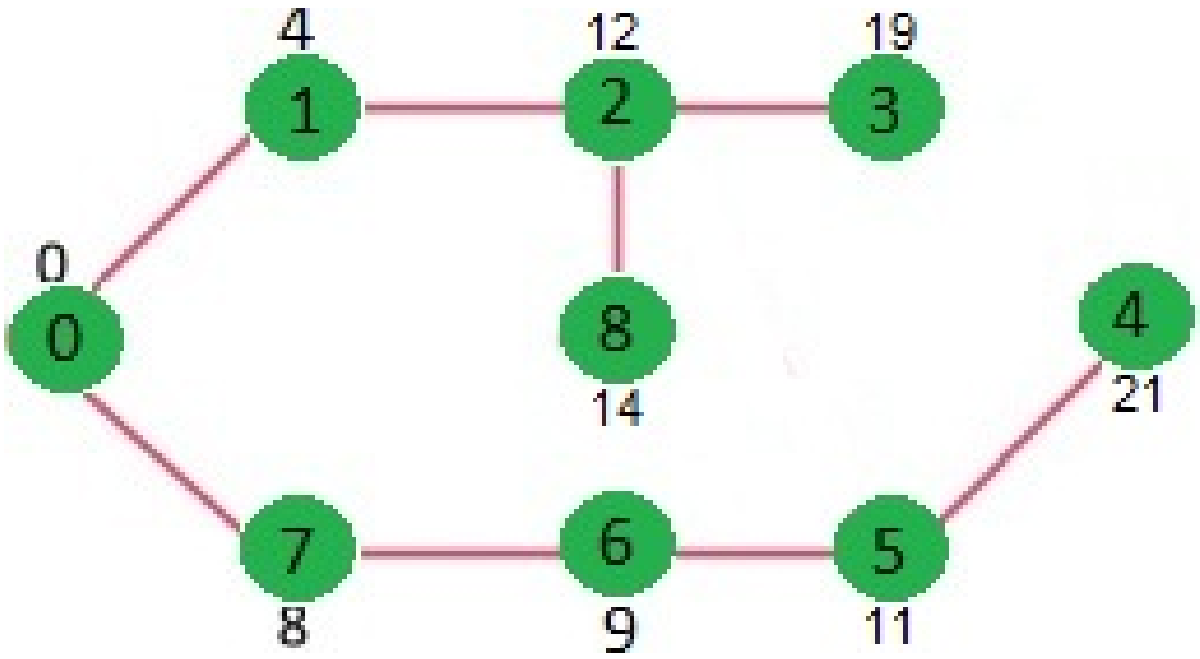

Figure 2: SP from source '0'