ISSN 2278-8808
An International Peer Reviewed
SJIF 2013 = 4.194
SCHOLARLY RESEARCH JOURNAL
FOR INTERDISCIPLINARY STUDIES

# AN INNOVATIVE TECHNIQUE IN DATA STRUCTURE TO FIND MINIMUM SPANNING TREE OF GRAPH

**Intedhar Shakir Nasir** (MSc computer science) Assistant lecturer, College of Medicine
University of Kerbala - IRAQ

***Abstract***

*Minimum spanning tree of any weighted graph is that tree whose sum of weights are least than any tree in the forest on that particular graph, because of this feature of minimum spanning tree, it can be applied to find the shortest route.The present study focuses that how data structure can be implemented to find the minimum spanning tree of any type of graph i.e. directed, undirected and weighted. This has been implemented inC programming language and the properties of graph implemented to find the final cost of tree. Adjacency matrix and adjacency list concept is used to organize the memory location of the graphs. Link list, tree and graph concept used to find minimum spanning. In the experiment review the result from KRUSKAL theorem and PRIMs theorem exactly same as the result got from the C source code.*

## 1. INTRODUCTION:

In computer science data structure is nothing but describing the data in various manner and performing different operations on data like traversing the data structure, adding records, deleting, records, modifying records, data structure provides the facility to store the data in linear and non linear form[1]. Linear and non linear are the basic type of the Data structure. Linear data structures stores the data in contiguous form where asNon linear data structure not stores the data in contiguous form. Data structure can be implemented in different programming languages like C, C++,JAVA& PASCAL. By using any programming language and data structure it is possible to solve any critical problem related to math, operation research and other

related subjects. By using same technique it is possible to manipulate all the operations that related to graph.

In data structure Tree is non linear data structure where data items can be stored and relationship can be established in hierarchical form. The present study relates to the minimum spanning tree of graph[2]. A graph data structure consists of a finite (and possibly mutable) set of nodes or vertices, together with a set of ordered pairs of these nodes (or, in some cases, a set of unordered pairs). These pairs are known as edges or arcs. As in mathematics, an edge (x,y) is said to point or go from x to y. The nodes may be part of the graph structure, or may be external entities represented by integer indices or references [7]. A graph data structure may also associate to each edge some edge value, such as a symbolic label or a numeric attribute [7].The minimum spanning tree for given a connected, undirected graph, a spanning tree of that graph is a subgraph that is a tree and connects all the verticestogether. A single graph can have many different spanning trees. We can also assign a weight to each edge, which is a number representing how unfavorable it is, and use this to assign a weight to a spanning tree by computing the sum of the weights of the edges in that spanning tree. A minimum spanning tree (MST) or minimum weight spanning tree is then a spanning tree with weight less than or equal to the weight of every other spanning tree. More generally, any undirected graph (not necessarily connected) has a minimum spanning forest, which is a union of minimum spanning trees for its connected components[6].

**2. OBJECTIVES OF THE STUDY:**

- To study the graph and its properties
- To study the various algorithm to find minimum spanning tree of the graph
- To Create memory organization for directed undirected and weighted type of graph
- To create source code in C using data structure to create graph, traversal of graph
- To analyze both algorithm with respect to output of source code

**3. PROPOSED METHODOLOGY:** The methodology involved the study of graph, properties of graph, memory organization of graph, source code in C using data structure, manual as well as experimental solution of Kruskal and PRIM algorithm.

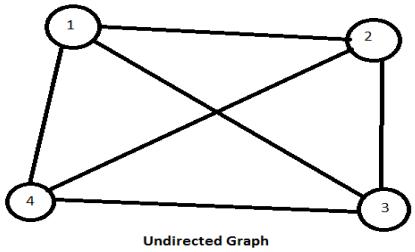
**Graph and its properties:** Graph is concept of Mathematics term which consists two terms i.e. vertices and edges. Each edge of graph is associated with unordered pair of vertices. A graph is a pair of Sets(V,E) where V is called the sets of vertices and E is called as the set of edges [3].

i.e. G=(V,E)

V(G) = (v0,v1…… vn) or set of vertices

E(G) = (e1,e2…….en) or set of edges.

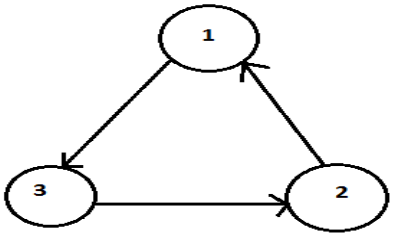
The graph may be directed and undirected type. The vertices and edges relation set for these type of graphs are as follows:



Undirected Graph

Here Set relations is : V(G) = {1,2,3,4}

E(G)={(1,2),(1,3),(1,4),(2,1),(2,2),(2,3),(3,1),(3,2),(3,4),(4,1),(4,2),(4,3)



Directed Graph

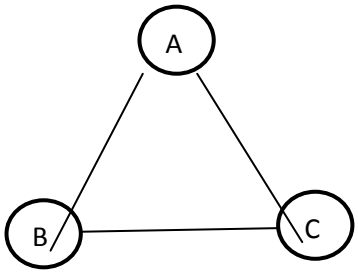
Here Set relation is

V(G) = {1,2,3}

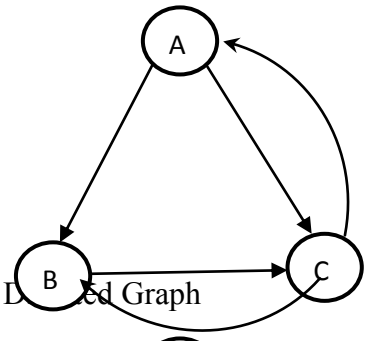
E(G)={(1,2),(1,3),(3,2)}

**4. MEMORY ORGANIZATION OF GRAPH:** To solve the minimum spanning tree using data structure using C programming it is necessary to design a structure of graph, then the components of graph i.e. nodes &edges and their weights. It is necessary to design the structure by considering the all properties of directed, undirected and weighted graph. There are two ways for representing the graph in computer memory i.e. Sequential and linked list representation. It

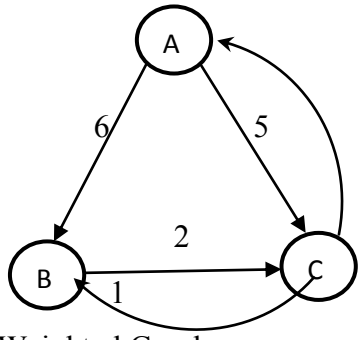
can be used as 1) adjacency matrix 2) adjacency list representation of memory organization by using adjacency matrix for directed, undirected and weighted graph as follows.




Undirected Graph




Directed Graph




Weighted Graph

Memory Organization using Adjacency matrix

| | A | B | C |
|---|---|---|---|
| A | 0 | 1 | 1 |
| B | 1 | 0 | 1 |
| C | 1 | 1 | 0 |

Undirected

| | A | B | C |
|---|---|---|---|
| A | 0 | 1 | 1 |
| B | 1 | 0 | 1 |

| | | | |
|---|---|---|---|
| C | 1 | 1 | 0 |

Weighted Graph

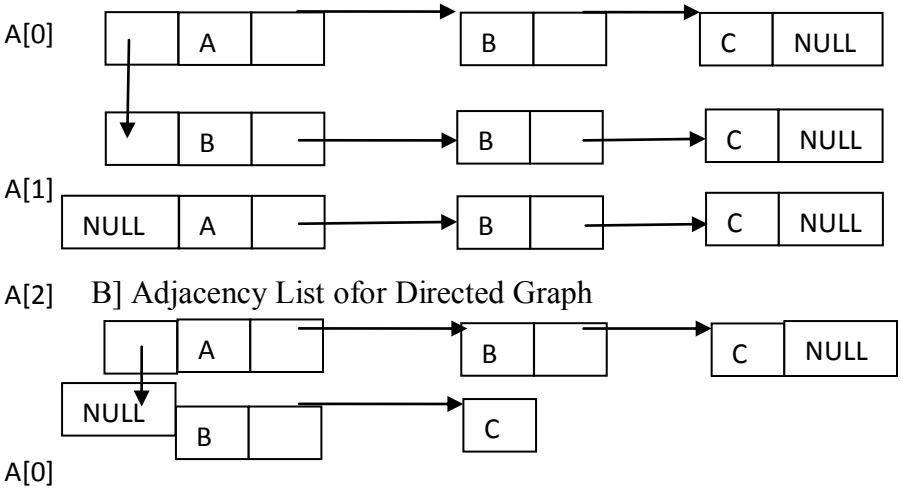
| | | | |
|---|---|---|---|
| A | B | C | |
| A | 0 | 1 | 1 |
| B | 1 | 0 | 1 |
| C | 1 | 1 | 0 |

Directed Graph

In the above memory organization, adjacency matrix for undirected graph is created by implying symmetric matrix i.e. for every I,j, A[i][j]= A[j][i] for directed graph, it is created by using the edge direction i.e. 1 for edge from 1 to j & 0 for no edge from I to j and for weighted it is created by using edge direction & placing weight value to I -> j. [4]

**4.1. Memory organization using adjacency List:**

A] Adjacency List for Undirected Graph




B] Adjacency List ofor Directed Graph




**5. MINIMUM SPANNING TREE**

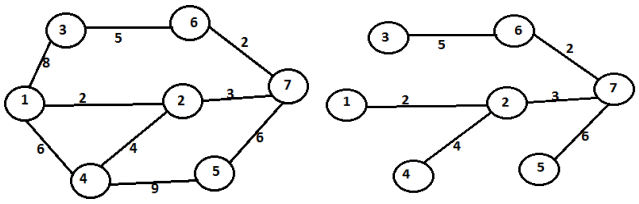
A[1] This is a method in which the sum of weights of the included edges is the minimum. There are various algorithm that helps to find minimum spanning tree. But most efficient and frequent algorithm to find minimum spanning tree are:

1] PRIM'S Algorithm 2] Kruskal Algorithm : Prim's algorithm is a greedy algorithm that finds a minimum spanning tree for a connected weightedundirected graph. This means it finds a subset

of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized [9]. NKruskal's algorithm is a greedy algorithm in graph theory that finds a minimum spanning tree for a connected weighted graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. If the graph is not connected, then it finds a minimum spanning forest (a minimum spanning tree for each connected component)[8]. The main difference between these two algorithms is that in Kruskal algorithm all edges are examined and one by one starting from the smallest edge. In PRIME'S algorithm it can be start with any node and add the other node in spanning tree on the basis of weight of edge connecting to that node.

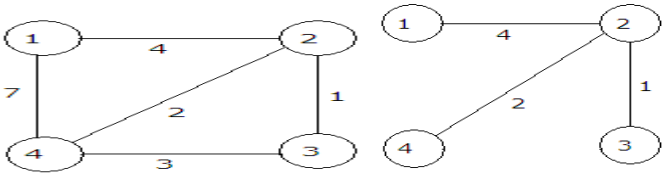
**6. PROGRAMMING LANGUAGE:** In present study to solve the minimum spanning tree C programming language is used. Using data structure, C source code concept i.e. structure, array and pointer the experiment have been performed and following experimental analysis found.

**7. EXPERIMENTAL REVIEW:** In experiment process following graphs with their weights and directions are entered through the C source code and observed the output with respect to Kruskal and PRIMs algorithm



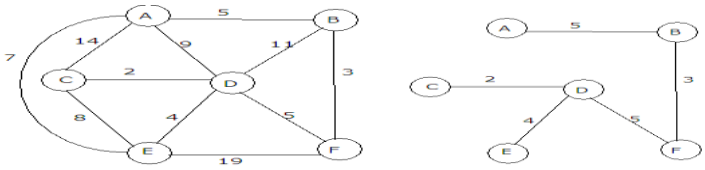

Minimum Spanning by Using Kruskal& PRIM

Minimum spanning tree by C source code




Minimum Spanning by Using Kruskal& PRIM

Minimum spanning tree by C source code



Minimum Spanning by Using Kruskal& PRIM

Minimum spanning tree by C source code

**8. CONCLUSION:** As the Minimum spanning tree is the tree in the graph which connects all the nodes in the graph without maintaining any cycle and the sum of weights of all the edges are least than any other tree in the graph. Therefore this technique can be implemented to Solve the PERT and CPM and transportation problem in operation research.From the experiment, it is observed that if the direction and edge weight of graph entered correctly the output from the experiment is exactly equal to the manual cost of the minimum spanning tree of the graph using Kruskal and PRIMs algorithm. The use of this can be implanted in Discrete Mathematic to solve the graphs problem.

**REFERENCES:**

Mathematical foundation of computer science by G. Shankar Rao

Discrete Mathematics by M.D. Bhagat& R.S. Bhamare

Data Structure using C by Aaron M. Tenenbaum, YedidyahLangsam, Moshe J. Augenstrin

Data Structure through C by YeshwantKanetkar

Data Structures through C in Depth. BPB publications by S.K. Srivastava and DeepaliSrivastava.

Pettie, Seth; Ramachandran, Vijaya (2002), "An optimal minimum spanning tree algorithm", Journal of the Association for Computing Machinery 49 (1): 16–34,doi:10.1145/505241.505243, MR 2148431.

Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001). Introduction to Algorithms (2nd ed.). MIT Press and McGraw–Hill. ISBN 0-262-53196-8.

Kruskal, J. B. (1956). "On the shortest spanning subtree of a graph and the traveling salesman problem". Proceedings of the American Mathematical Society 7: 48–50. doi:10.1090/S0002-9939-1956-0078686-7. JSTOR 2033241.

R. C. Prim: Shortest connection networks and some generalizations. In: Bell System Technical Journal, 36 (1957), pp. 1389–1401