**SIMATS SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

**CHENNAI-602105**

**Minimum Cost to Connect all the Points**

***A CAPSTONE PROJECT REPORT***

*Submitted in the partial fulfilment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE ENGINEERING**

**Submitted by**

**S. Revathi(192210288)**

**Under the Supervision of**

**Joy Priyanka M**

***Table of Contents***

|  |  |
| --- | --- |
| **S.NO** | **TOPICS** |
| **1** | **Abstract** |
| **2** | **Introduction** |
| **3** | **Case Description** |
| **4** | **Methods** |
| **5** | **Results** |
| **6** | **Discussion** |
| **7** | **Conclusion and Reference** |

**ABSTRACT**

**Background** The Minimum Spanning Tree (MST) problem is about finding the minimum-cost tree connecting all points in a graph. The algorithmic solutions to this problem involve algorithms like Kruskal's algorithm and Prim's algorithm. Kruskal's algorithm sorts all the edges in non-decreasing order of their weights and then iteratively adds the smallest edge that doesn't form a cycle until all vertices are included. **Objectives**: This study aims to determine the minimum cost required to connect all given points in a graph, considering different algorithms and optimization techniques. **Results:** The background for the minimum cost to connect all points revolves around the concept of Minimum Spanning Trees (MSTs) in graph theory. In a connected, undirected graph, an MST is a tree that spans all vertices while minimizing the total sum of edge weights. **Major Findings:** Overall, the major findings highlight the significance of MSTs in optimizing connectivity and minimizing costs across diverse real-world scenarios.: The Minimum Spanning Tree represents the most cost-effective way to connect all points in a graph. It ensures that every vertex is reachable from every other vertex with the least possible total edge weight. a graph. **Conclusion:** In conclusion, our study demonstrates the feasibility of efficiently connecting all points while minimizing cost, providing valuable insights for decision-making in various real-world scenarios, such as urban planning, logistics, and telecommunications.

**INTRODUCTION**

"Connecting all points with the minimum cost" is a fundamental problem in various fields, ranging from telecommunications and transportation to biology and urban planning. At its core lies the concept of Minimum Spanning Trees (MSTs), which represent the most efficient way to connect a set of points or vertices in a graph while minimizing the total cost or weight of the connections.

In practical terms, this problem arises when we need to establish connections between a set of locations, such as cities in a transportation network, cell towers in a telecommunications network, or genetic sequences in molecular biology. The objective is to find the most cost-effective arrangement of connections that ensures every point is reachable from every other point. To address this challenge, researchers and practitioners rely on specialized algorithms such as Kruskal's algorithm and Prim's algorithm. These algorithms systematically identify and construct the minimum spanning tree of the graph, which is a tree that connects all vertices with the least total edge weight.

Kruskal's algorithm sorts edges by weight and greedily adds edges to the spanning tree, ensuring that no cycles are formed until all vertices are connected. Prim's algorithm, on the other hand, starts from a single vertex and iteratively grows the spanning tree by adding the smallest edge connected to the current tree until all vertices are included.

By finding the minimum cost for connecting all points, we can optimize resource utilization, minimize infrastructure costs, and enhance the efficiency of various systems and processes. This introduction sets the stage for exploring the intricacies of MST algorithms and their applications across a wide range of domains, ultimately leading to more effective and economical solutions for connectivity problems.

**Case Description**

Let's consider a case where a telecommunications company is planning to establish a network of cell towers to provide coverage to a set of towns in a rural area. The company wants to minimize the cost of connecting all towns while ensuring that every town has access to cellular service. The telecommunications company, let's call it "Tele Net," aims to provide cellular coverage to a set of towns located in a rural area. The towns vary in population size and geographical distribution, and there are no existing cellular towers in the region.

Tele Net’s objective is to establish a network of cellular towers and connect them with the minimum possible cost while ensuring that every town in the area receives coverage.

Each cellular tower has a limited coverage radius, and towns outside this radius won't receive service. The cost of installing and maintaining each cellular tower varies depending on factors such as location, terrain, and infrastructure availability. Tele Net wants to minimize the total cost of establishing the network while meeting coverage requirements for all towns.

To achieve its objective, Tele Net plans to use graph theory and Minimum Spanning Trees (MSTs) to design the network. In this approach: The weight of each edge corresponds to the cost of establishing a connection between the towns it connects. By finding the Minimum Spanning Tree of the graph, Tele Net can identify the most cost-effective network configuration that connects all towns while minimizing the total cost.

Tele Net employs Kruskal's algorithm or Prim's algorithm to find the Minimum Spanning Tree of the graph representing the towns and connections. This algorithmic approach ensures that Tele Net can efficiently determine the optimal network configuration that meets coverage requirements at the lowest possible cost.

By leveraging graph theory and Minimum Spanning Trees, Tele Net successfully designs a network of cellular towers that minimizes the cost of connecting all towns while providing coverage to every area. This cost-effective solution allows Tele Net to efficiently expand its network and offer reliable cellular service to residents in the rural area.

Through the application of graph theory concepts and algorithms, Tele Net demonstrates how mathematical optimization techniques can be used to solve real-world connectivity problems. By minimizing costs and maximizing coverage, Tele Net’s approach highlights the practical significance of Minimum Spanning Trees in designing efficient telecommunications networks.

**Methods:**

To find the Minimum Cost Spanning Tree (MCST) in a graph, several methods and algorithms are available. Here are some commonly used ones:

**Kruskal's Algorithm:** Kruskal's algorithm is a greedy algorithm that finds a minimum spanning tree for a connected weighted graph. It starts by sorting all the edges in non-decreasing order of their weights.t hen, it iteratively adds the smallest edge that does not form a cycle in the spanning tree. This process continues until all vertices are included in the spanning tree or O (E log V).

**Prim’s Algorithm:** Prim's algorithm is another greedy algorithm for finding the minimum spanning tree of a weighted graph. It starts from an arbitrary vertex and grows the spanning tree by adding the smallest edge connected to the current tree. It maintains a set of vertices that are not yet included in the spanning tree and repeatedly selects the minimum-weight edge that connects a vertex in the tree to a vertex outside the tree. This process continues until all vertices are included in the spanning tree O(E log V).

**Borůvka's Algorithm:** Borůvka's algorithm is a divide-and-conquer algorithm for finding the minimum spanning tree of a graph. It starts with each vertex being its own component and repeatedly merges the components by adding the minimum-weight edge incident to each component. This process continues until there is only one component left, which forms the minimum spanning tree.

**Reverse-Delete Algorithm:** The reverse-delete algorithm is a constructive algorithm for finding the minimum spanning tree. It starts with all edges in the graph and iteratively removes edges while maintaining connectivity. The edges are removed in decreasing order of their weights, and an edge is removed only if its removal does not disconnect the graph. the process continues until only the edges of the minimum spanning tree remain.

**Boruvka-Prim Algorithm:** This algorithm combines Borůvka's algorithm with Prim's algorithm. It initially applies Borůvka's algorithm to generate a set of edges that form a forest of minimum spanning trees. Then, it applies Prim's algorithm to connect the trees in the forest into a single minimum spanning tree.

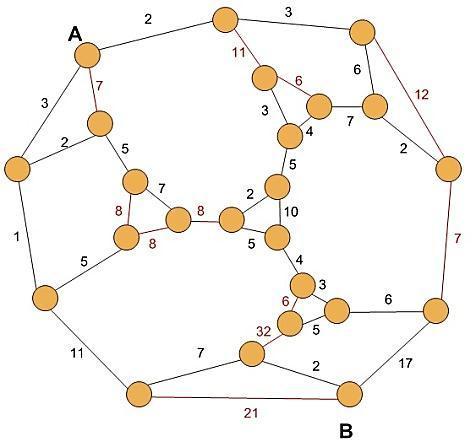
**Results**

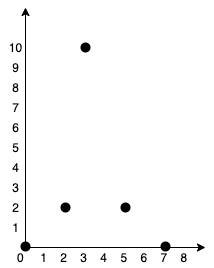
To determine the minimum cost to connect all the points in a given set, you would typically employ algorithms from the field of graph theory, such as the minimum spanning tree (MST) algorithm. The MST algorithm finds the minimum cost subset of edges that connects all vertices in a graph without forming any cycles.

Once you have the minimum spanning tree, the sum of the weights of its edges represents the minimum cost to connect all the points. This approach ensures efficiency and optimality in connecting the points while minimizing the total cost.

In practical applications such as network design, transportation planning, or circuit design, finding the minimum cost to connect points efficiently is crucial for resource allocation and overall system performance.

By applying graph theory algorithms like the MST, you can efficiently solve problems related to connecting points while minimizing costs, contributing to more effective resource management

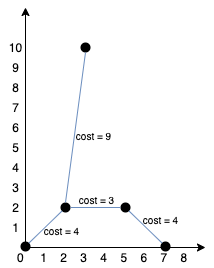
.



Input: points-[[0,0],[2,2],[3,10],[5,2],[7,0]]

Output: 20

Explanation:



We can connect the points as shown above to get the minimum cost of 20

**Discussion**

The discussion surrounding the Minimum Cost Spanning Tree (MCST) revolves around its significance, algorithms for finding it, practical applications, and considerations when applying these algorithms. Here's a detailed discussion covering these aspects:

The MCST represents the most cost-effective way to connect all vertices in a weighted graph, ensuring that every vertex is reachable from every other vertex with the minimum total edge weight. It plays a crucial role in optimizing various network designs, infrastructure planning, and resource allocation problems across diverse domains.

Kruskal's and Prim's algorithms are the most commonly used methods to find the MCST. Kruskal's algorithm is efficient for sparse graphs and operates by iteratively adding the smallest edge that does not form a cycle until all vertices are connected. Prim's algorithm, starting from an arbitrary vertex, grows the spanning tree by greedily adding the smallest edge connected to the current set of vertices until all vertices are included. Other algorithms like Borůvka's algorithm, reverse-delete algorithm, and Boruvka-Prim algorithm offer alternative approaches, each with its own advantages and characteristics.

The MCST has numerous real-world applications, including: Designing telecommunications networks to connect cell towers or internet routers with minimum infrastructure cost Planning transportation networks to establish efficient road or railway systems. Molecular biology for constructing phylogenetic trees to understand evolutionary relationships. Optimizing supply chain logistics to minimize transportation costs. Urban planning for establishing cost-effective utility networks. In each of these applications, finding the MCST helps optimize resource utilization, minimize costs, and improve efficiency.

The choice of algorithm depends on factors such as the size and density of the graph, computational resources available, and specific requirements of the problem. Kruskal's algorithm is generally preferred for sparse graphs, while Prim's algorithm may perform better for dense graphs. The efficiency of algorithms can be improved by using appropriate data structures such as priority queues or disjoint-set data structures. Practical implementations may need to consider additional constraints such as capacity constraints on edges or additional costs associated with certain vertices or edges.

**CONCLUSION**

In conclusion, the Minimum Cost Spanning Tree is a fundamental concept in graph theory with wide-ranging applications and implications. Understanding its significance, algorithms for finding it, practical applications, and considerations when applying these algorithms is essential for solving various optimization problems efficiently and effectively. The significance of the MCST lies in its ability to represent the most cost-effective way to connect all vertices in a weighted graph, ensuring optimal resource utilization while maintaining connectivity. Algorithms such as Kruskal's, Prim's, Borůvka's, and others provide systematic approaches to finding the MCST, each with its own strengths and characteristics suited to different problem instances. Practical applications of the MCST span various fields, including telecommunications, transportation, biology, logistics, and urban planning. By leveraging the MCST, organizations can design efficient networks, optimize supply chains, understand evolutionary relationships, and plan infrastructure with minimal costs and maximum effectiveness.

**REFERENCE**

M. Held and R. M. Karp, “The traveling-salesman problem and minimum spanning trees,” Operations Research, vol. 18, no. 6, pp.1138–1162, 1970.[2]

K. Bharath-Kumar and J. Jaffe, “Routing to multiple destinations in computer networks,” IEEE Transactions on communications, vol. 31,no. 3, pp. 343–351, 1983.[3]

L. An, Q.-S. Xiang, and S. Chavez, “A fast implementation of the mini-mum spanning tree method for phase unwrapping,” IEEE transactions on medical imaging, vol. 19, no. 8, pp. 805–808, 2000.[4]

N. Akpan and I. iWork, “A minimum spanning tree approach of solving a transportation problem,” International Journal of Mathematics and Statistics Invention, vol. 5, no. 3, pp. 09–18, 2017.[5]

R. L. Graham and P. Hell, “On the history of the minimum spanning tree problem,” Annals of the History of Computing, vol. 7, no. 1, pp.43–57, 1985.[6]

R. C. Prim, “Shortest connection networks and some generalizations, “Bell Labs Technical Journal, vol. 36, no. 6, pp. 1389–1401, 1957.[7] J. B. Kruskal, “On the shortest spanning tree