C2

Date:- / /
Page No.:- 21

Title :- Minimum cost spanning Tree Algorithm for connecting multiple offices.

problem statement :- You have a business with several offices; you want to lease phone lines to connect them upwith each other; and the phone company charge. different amounts of money to connect diffrent pairs of cities you want a set of lines that connects all your offices with a minimum total cost Solve the problem by suggesting appropriate data structure

objectives :- To find a minimum cost spanning tree that connects all the offices together

Software Requirement :-

Theory :- This problem can be solved using the minimum spanning tree algorithm, which is graph theory algorithm that finds a tree that connects all the vertices of a graph and has the minimum possible total edge weight To apply the minimum spanning tree algorithm we can represent the network of offices and phone line connections as an undirected graph, where each office is a vertex and each phone line is an edge with a height representing the cost of the connection we can use an adjacency matrix or an adjacency list to represent the graph An adjacency matrix is a two dimensional array where the rows and columns represent the vertices and the entries represent of linked lists where each vertex has a list of it's adjacent vertices and their edges weights

once we have the graph representation, we can apply the minimum spanning tree algorithm. There are two commonly used algorithms for finding the minimum spanning tree.

1] Kruskal's algorithm
2] prim's algorithm

Kruskal's algorithm sorts the edges in asending order by weight, and then adds each edge to the minimum spanning tree as long as it does not create a cycle. This algorithm can be implemented using a disjoint-set data structure to efficiently detect and keep track of which vertices are connected

prim's algorithm starts with an arbitary vertex and greedily adds the cheapest edge that connects a vertex not already in the tree to the tree this algorithm can be implemented using priority queue to efficiently find the cheapest edge to add to the tree at each step

The problem of finding the minimum spanning tree is a classic problem in graph theory, and it has numerous applications in various fields, including computer networks, transportation and biology.

To apply the minimum spanning tree algorithm to the problem of leasing phone lines, we can represent the network of offices and phone line connections as an undirected graph, where each office is a vertex and each phone line is an edge with a weight representing the cost of the connection

once we have represented the graph, we can apply the minimum spanning tree algorithm to find the set of the Phone lines that connects all the offices with a minimum

total cost Both kruskal's of o(elogе), where 'e' is no of edges in the graph

Algorithm :-
- Here we can see the kruskal's algorithm which is a greedy algorithm that finds a minimum spanning tree for a connected weighted graph
- Here we need a data structure to store the edges and another data structure to keep track of the connected components of the graph

Here is the algorithm is ~~sp~~ pseudocode
1. Sort all the edges in increasing order of their weights.
2. Initialize an empty set of edges and an empty union find data structure with one set for each vertex
3. for each edge in the sorted list of edges a. If the edge connects two different sets in the union data structure add it to the set of edges and merge the sets
4. Return the set of edges

The time complexity is o(elogе), where e is the number of edges is the graph, due to cost of sorting the edges and performing the union-find operations
The space complexity is o(V+E), where V is the number of the number of vertices for storing the edges and the union find data structure

Conclusion :- The problem of connecting multiple offices ~~dfull~~ with a minimum total cost can be solved by representing the graph using an appropriate data structure such as an adjacency matrix or an adjacency list and the applying a suitable MST algorithm