BILKENT UNIVERSITY CS 202 HOMEWORK 4 REPORT

NAME: Simay Uygur **ID:** 22203328
SECTION: 1 **DATE:** 30.12.2024

Question 1

As it is requested to find the minimum spanning tree, we must find the minimum weight edges and add those to our tree, but while adding, we shouldn't repeat the same vertices as they have the minimum edge among others. In the question, it is given that 10 vertices start from 1 to 10 and the graph is complete, meaning every vertex is connected to every other vertex. As those edges are weighted according to the formula given, the first few edges that is 1 2, or 3 have different weights among each other, but the rest of the vertices have their edge weights in a way that both of the vertices' values are added. As can be seen below, the graph is complete and we can choose any way to span the graph, but the minimum spanning tree is our concern.

According to **Kruskal's algorithm**, we should start from the edge with minimum weight, therefore in this problem, it is the best fit to use this algorithm.

We start from the edge (1,2)

The edges selected for the MST:

(1,2) -> weigth 5 (w=i^2+j^2 , i+j<5)

(1,4)-> weigth 5 (w=i+j , i+j>=5)

(2,3)-> weigth 5 (w=i+j , i+j>=5)

(1,5)-> weigth 6 (w=i+j , i+j>=5)

(1,6)-> weigth 7 (w=i+j , i+j>=5)

(1,7)-> weigth 8 (w=i+j , i+j>=5)

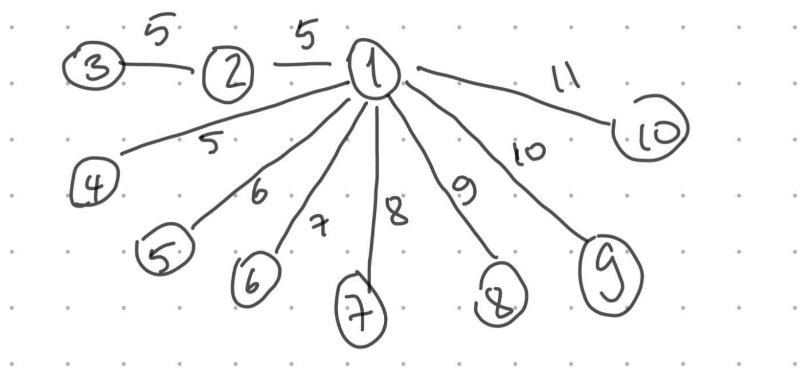
(1,8)-> weigth 9 (w=i+j , i+j>=5)

(1,9)-> weigth 10 (w=i+j , i+j>=5)

(1,10)-> weigth 11 (w=i+j , i+j>=5)

5+5+5+6+7+8+9+10+11 = 66 is the total weight of the minimum spanning tree.

As there should be 9 edges (N-1), and 10 vertices (N) in the minimum spanning tree, the final form would be like below. There are no cycles in the final form because it is a tree.



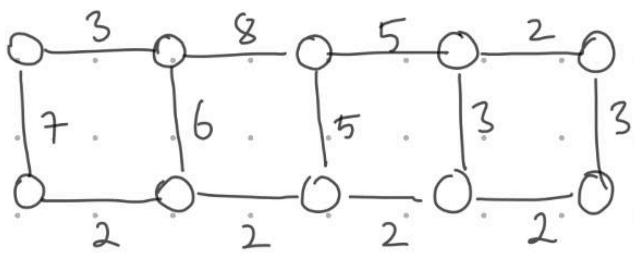
Question 2

In this problem the graph is sparse and has relatively few edges (13 edges and 10 vertices) therefore, **Kruskal's algorithm is more efficient** than Prim's algorithm. As we are dealing an adjacency matrix in Prim's algortihm its time complexity is O(V^2) but, for Kruskal's algorithm its time complexit is O(ElogN) because it uses adjacency matrix to add dependencies of the current element and uses a heap.

As our goal is minimizing the cost of LAN wiring, we should create a minimum spanning tree which there is no cycles and there is a path to all elements. All rooms are connected in that spanning tree because every element should be included as its definition contains spanning.

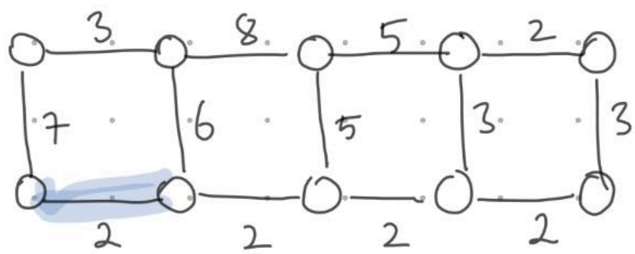
So in Kruskal's algorithm, all of the edges of the graph must be sorted according to their weights and they should be in ascending order as a first step. Then by this sorted edges, processing edges one another, it adds the minimum wight edge among the remaining ones and adds to the tree. But if that edge forms a cycle in the tree, it deforms the MST property, therefıre it is not added. This process is called union-find structure. Also Kruskal's algorithm processes edge by edge that means it is edge-centric.

The steps of this algorithm is below.

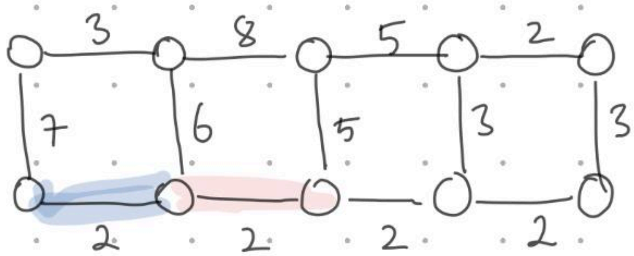


Step 1: selecting the minimum weight edge (as there are too many of them we will choose arbitrarily as long as they don't form a cycle)

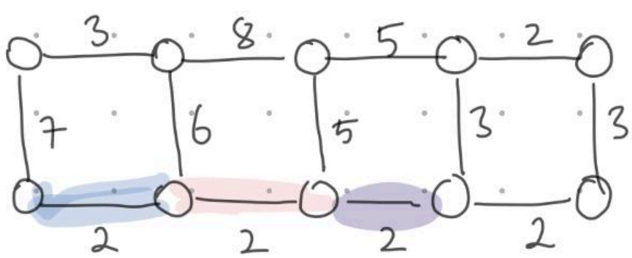
step



1

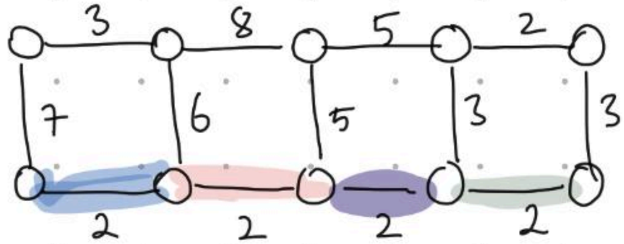


2

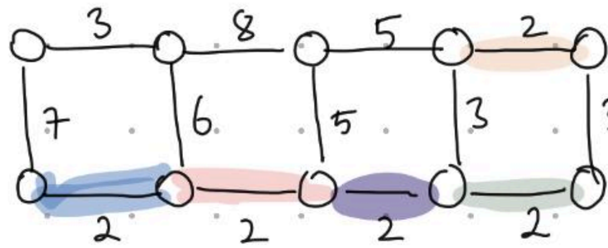


3

Step

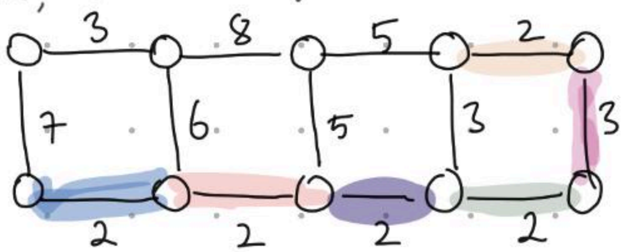


4

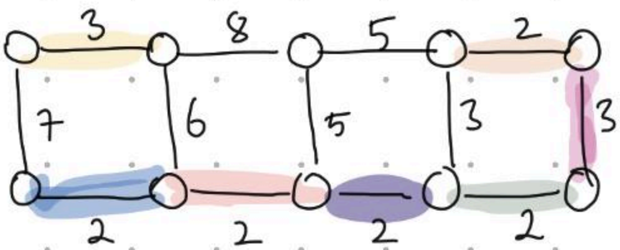


5

now, minimum weights → 3 (2 edges have the same weight)

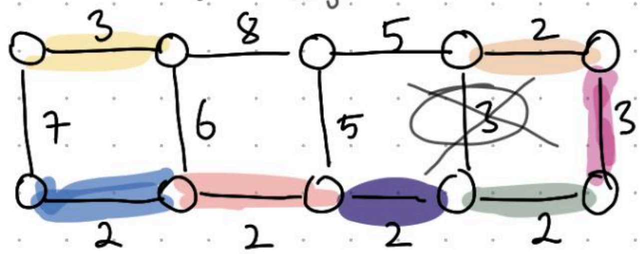


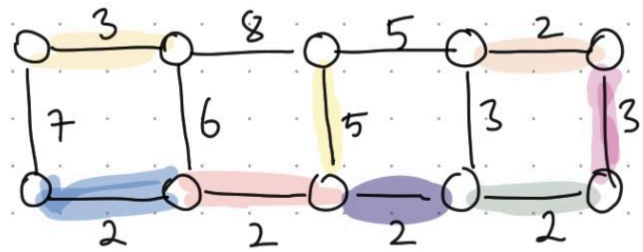
6



7

third 3 weighted edge can't be selected because it will form a cycle

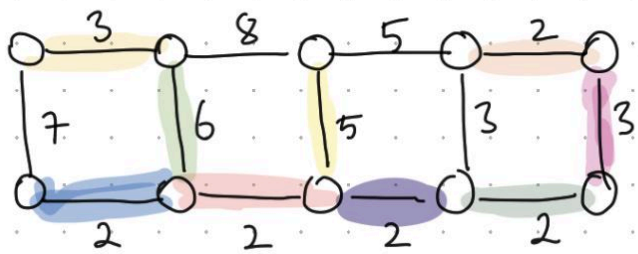




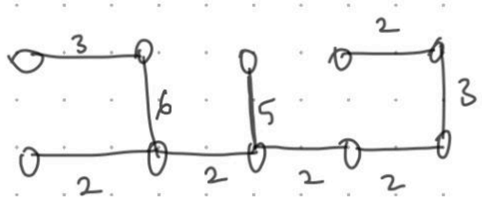
any 5 can be selected (it doesn't matter)

Step 8

Step 9



every vertex is connected, minimum spanning tree is formed



→ it may have different forms because of the repeated 3's and 5's.

But in the final form, all edges with weigh 2 must be in the MST, one of the 3 edge should be excluded as they form a circle, one of the 5 edge shouldn't be selected as they go to the same vertex.

The final minimum spanning tree's edges are
2+2+2+2+2+3+3+5+6 = 27 is the weight of the minimum spanning tree of this problem.