Pak-Austria Fachhochschule: Institute of Applied Sciences & Technology, Haripur, Pakistan

School of Computing Science

DSA

Submitted By: Haider Nasir

Registration # B24S0308AI088

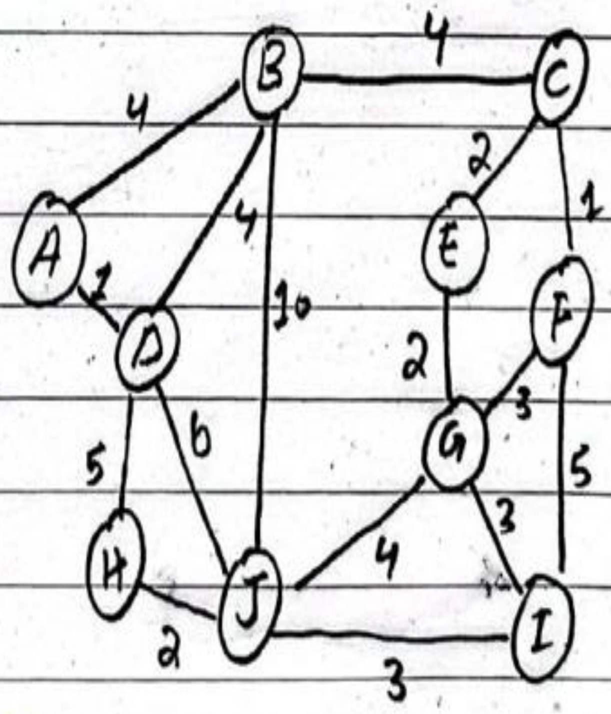
Submitted to: Sir Aamir Malik

Date of submission: 05/1/2025

Question No.1: Kruskal Algorithms.

This algorithm create a forest of trees. Initially the forest consists of n single node trees (and no edges). At each step, we add one edge (the cheapest one) so that it joins two trees together. If it were to form a cycle, it would simply link two nodes that were already part of a single connected tree, so that this edge would not be needed.

Complete Graph.



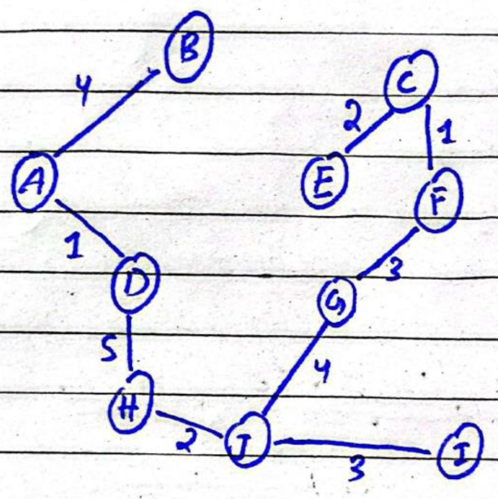
Answer:

According to the krustkal algorithen we will make tree with help of those point whose weight is less.

- Edges:

A-D(1), C-F(1), E-C(2), E-G(2), H-J(2), F-G(3), G-I(3), I-J(3), A-B(4), B-D(4), B-C(4), G-J(4), D-H(5), F-I(5), D-J(6), B-G(10).

Among these edges we take the edge with the minimum weight. The edges are A-D, C-F, E-C, E-G, H-J, F-G, G-I, A-B, these were the points that will cover all the vertix with the minimum weight.



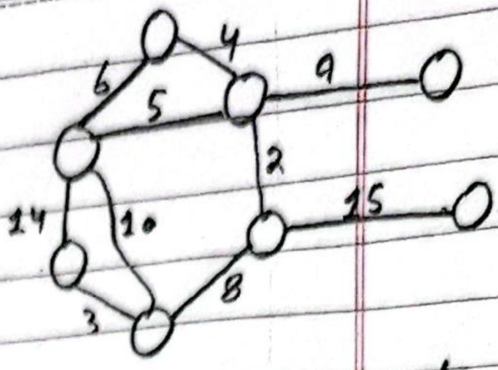
This is the minimum spaining tree covering all the points with the minimum weight i.e: 25.

Question No. 2:

Prim's algorithms.

Date:

M T W T F S



Run on example graph.

MST-Prim(G, w, r)
    Q = V[G];
    for each u ∈ Q
        key[u] = ∞;
    key[r] = 0;
    p[r] = NULL;
    while (Q not empty)
        u = ExtractMin(Q);
        for each v ∈ Adj[u]
            if (v ∈ Q and w(u,v) < key[v])
                P[v] = u;
                key[v] = w(u,v);

Run Prims algorithm on above example and write the answer for above example

Answer:

In this prime algorithm we first need to pick the node. So we will make the minimum spainning tree (MST).

Now we first consider bottom-left node as A. key[A] = 0, P[A] = NULL and the other keys are key[others] = ∞. Now we will extract Min(Q) → Node A. A node is connected with 3 the key update to 3, parent = A. Node 14 also connected to A key update to 14 parent = A. Then the third node 10 connected to A

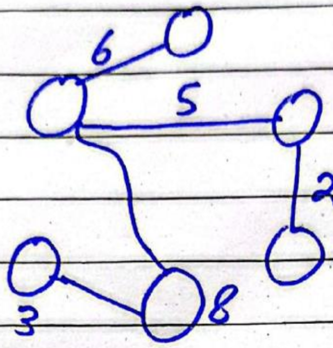
Date:

M T W T F S

Key updated to 10, parent = A. Now from these we pick node with minimum weight i.e Min Key = 3. Now we call it as B. The B node edge is connected to edge with weight 8 now we update over key to 8, parent = B. Now we consider Min Key = 8. Now we call it as C. The C node edge is connected to edge with weight 2. Now update key to 2, parent = C. C is also connected with edge whose weight is 15. Update key to 15, parent = C. Now the Min key = 2. Now we call it as D. The D node edge is connected to edge with weight 4 now we update over key to 4, parent = D. Similarly the edge is also connected with the weight 9, now we update over key to 9, parent = D.

Min key = 4 now we call it as E. E is connected to the edge whose weight is 5 update key to 5, parent = E. Now Min key = 5 now we call it as F. F is no more further connected to any smaller edge. After F their is an other edge whose weight is 9 we pick it and call it as G.

MST edges are:
(6,5), (6,10), (5,4), (4,2)
(2,8), (8,3), (3,10). These all vertices will make the MST.



Thus, we make the MST with minimum edges.