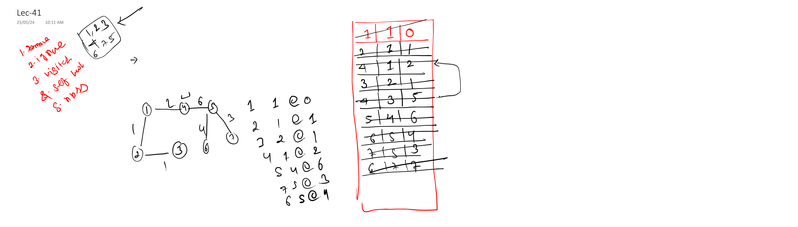
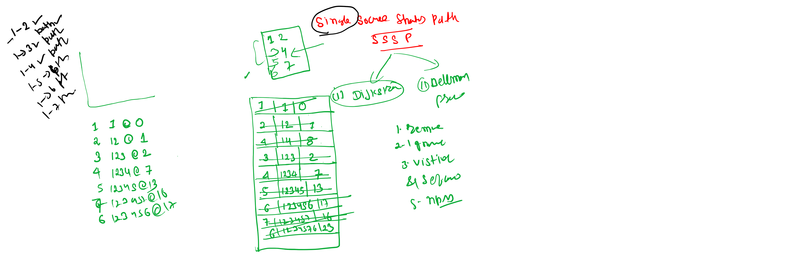
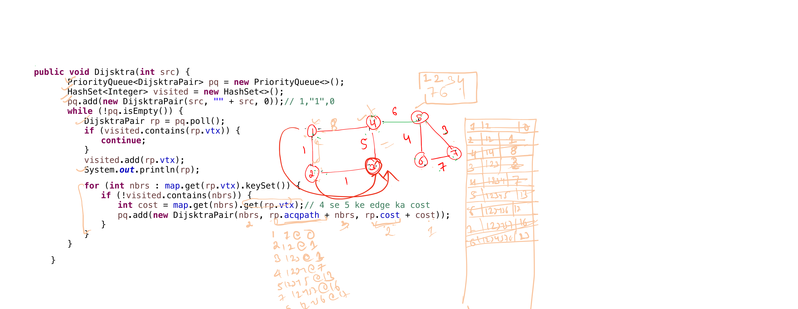
Lec-41

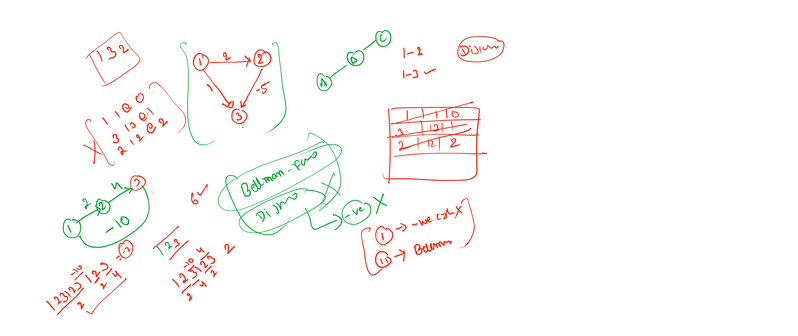


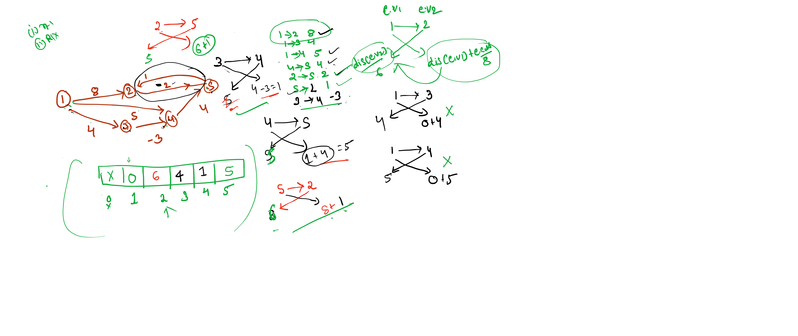


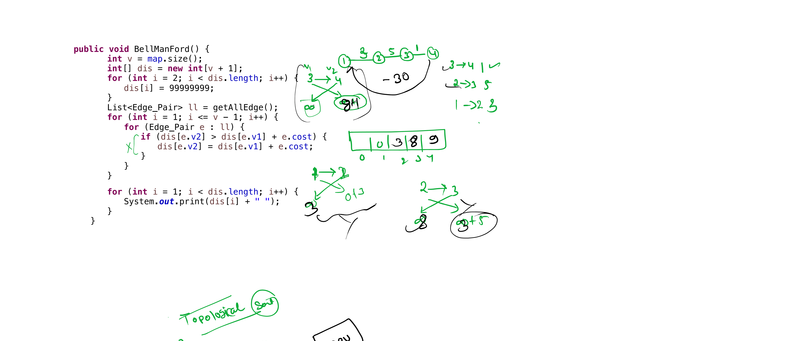


```
public void Dijsktra(int src) {
        PriorityQueue<DijsktraPair> pq = new PriorityQueue<>();
        HashSet<Integer> visited = new HashSet<>();
        pq.add(new DijsktraPair(src, "" + src, 0));// 1,"1",0
        while (!pq.isEmpty()) {
            DijsktraPair rp = pq.poll();
            if (visited.contains(rp.vtx)) {
                continue;
            }
            visited.add(rp.vtx);
            System.out.println(rp);

            for (int nbrs : map.get(rp.vtx).keySet()) {
                if (!visited.contains(nbrs)) {
                    int cost = map.get(nbrs).get(rp.vtx);// 4 se 5 ke edge ka cost
                    pq.add(new DijsktraPair(nbrs, rp.acqpath + nbrs, rp.cost + cost));
                }
            }
        }
    }
```

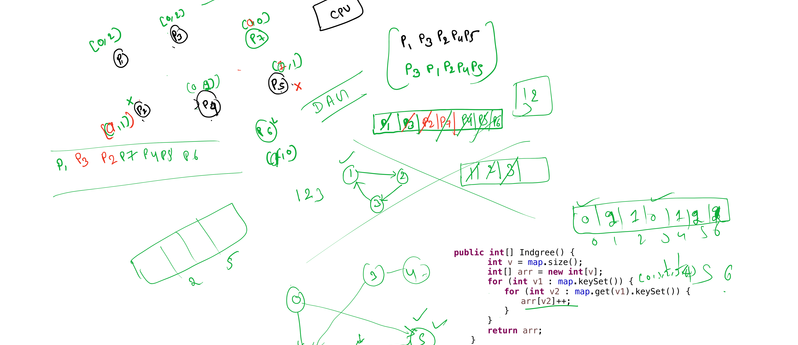


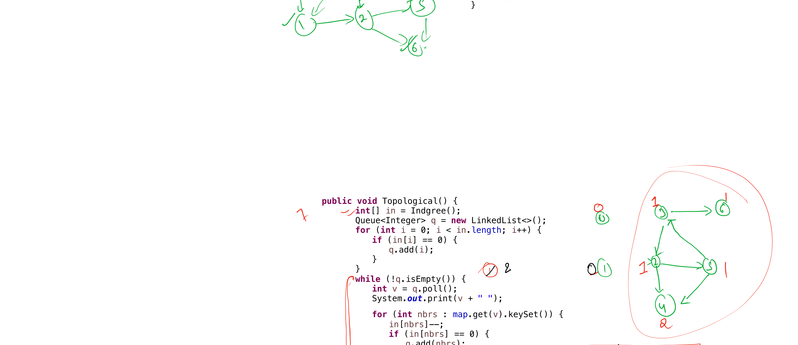




```
public void BellManFord() {
        int v = map.size();
        int[] dis = new int[v + 1];
        for (int i = 2; i < dis.length; i++) {
            dis[i] = 99999999;
        }
        List<Edge_Pair> ll = getAllEdge();
        for (int i = 1; i <= v - 1; i++) {
            for (Edge_Pair e : ll) {
                if (dis[e.v2] > dis[e.v1] + e.cost) {
                    dis[e.v2] = dis[e.v1] + e.cost;
                }
            }
        }

        for (int i = 1; i < dis.length; i++) {
            System.out.print(dis[i] + " ");
        }
    }
```





```
public int[] Indgree() {
        int v = map.size();
        int[] arr = new int[v];
        for (int v1 : map.keySet()) {
            for (int v2 : map.get(v1).keySet()) {
                arr[v2]++;
            }
        }
        return arr;
    }
```

```
public void Topological() {
        int[] in = Indgree();
        Queue<Integer> q = new LinkedList<>();
        for (int i = 0; i < in.length; i++) {
            if (in[i] == 0) {
                q.add(i);
            }
        }
        while (!q.isEmpty()) {
            int v = q.poll();
            System.out.print(v + " ");

            for (int nbrs : map.get(v).keySet()) {
                in[nbrs]--;
                if (in[nbrs] == 0) {
                    q.add(nbrs);
                }
            }
        }
    }
```