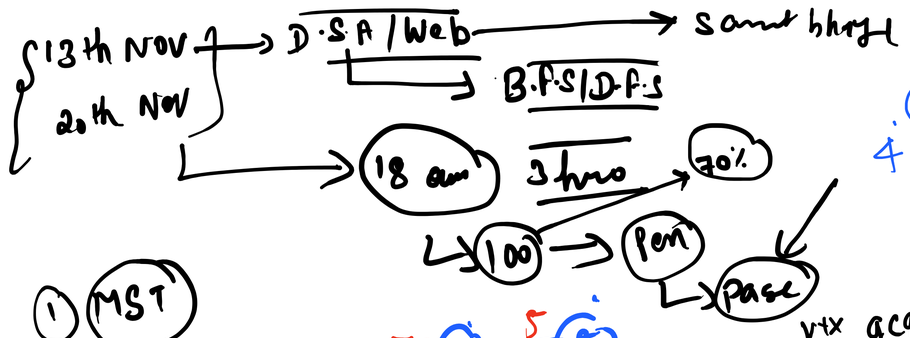
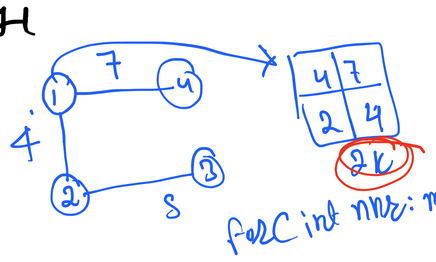
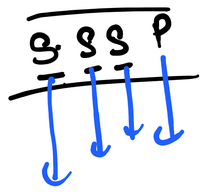
Lec-57

12/11/25 10:19 AM

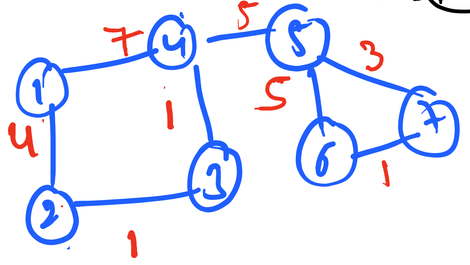




for(int nbr: map.get(ep.vtx).keySet()) {
int cost = map.get(ep.vtx).get(nbr)

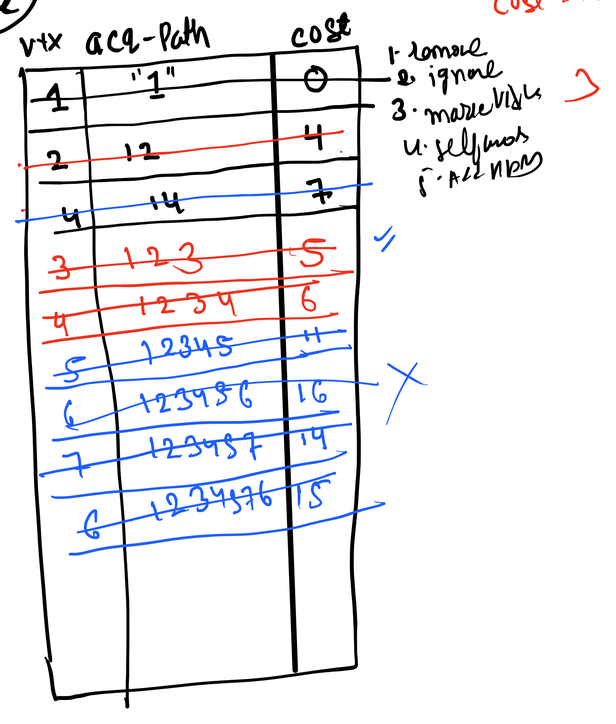


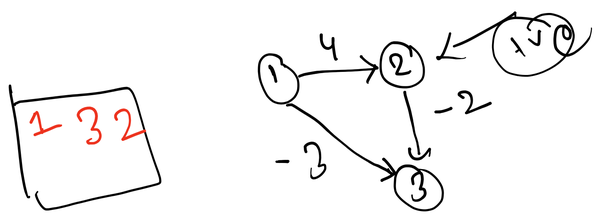
Dijkstra
Bellman ford

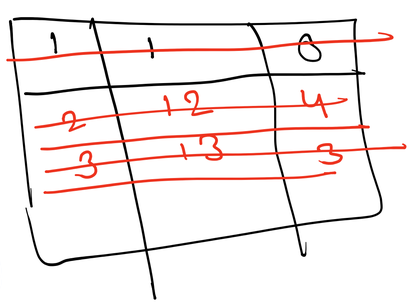


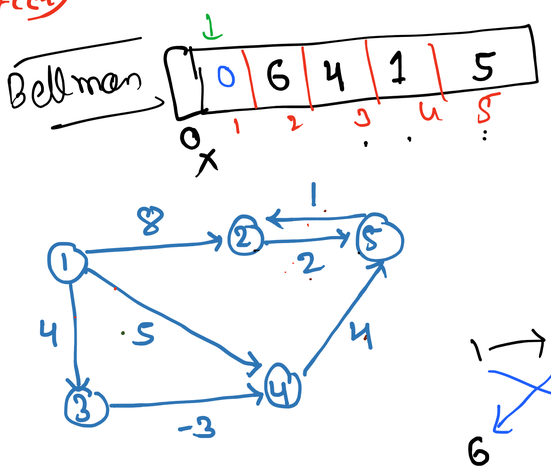
Visited
1 2 3 4
5 7 6

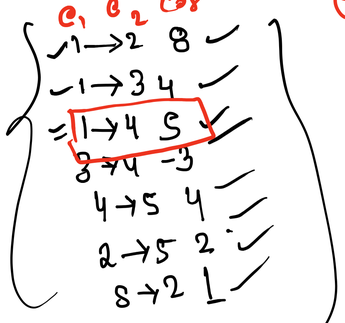
1 1 @ 0
2 12 @ 4
3 123 @ 5
4 1234 @ 6
5 12345 @ 11
7 123457 @ 14
6 1234576 @ 15

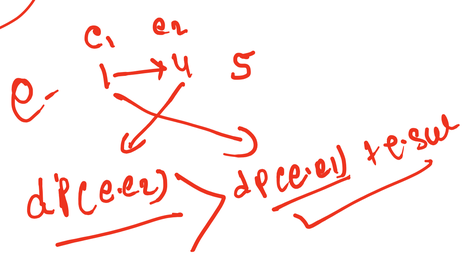


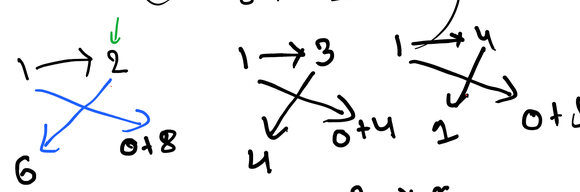


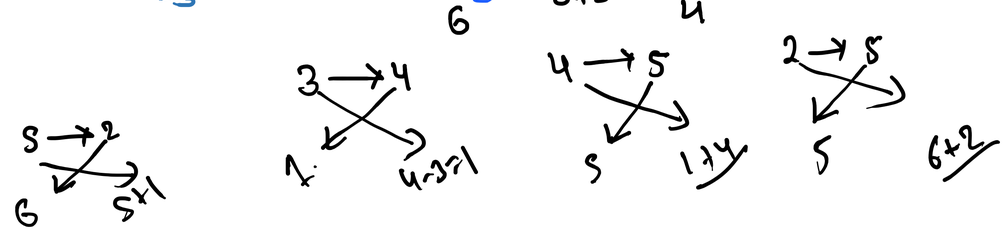










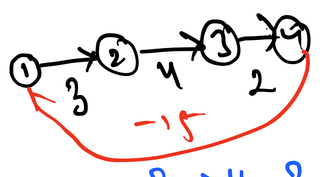


```
PriorityQueue<DijkstraPair> pq = new PriorityQueue<>((a,b)->a.cost-b.cost);
```

2c - 4 > 0

forcint e1: map.keyset()
forcint e2: map.get(e1).keyset()
cost = map.get(e1).get(e2)


```
public void BellManFord_Algo() {
      int v = map.size();
      int[] dp = new int[v + 1];
      // src ==1
      for (int i = 2; i < dp.length; i++) {
         dp[i] = 9999999;
      }
      List<EdgePair> ll = getAllEdge();
      for (int i = 1; i < v; i++) {
         for (EdgePair e : ll) {
            if (dp[e.e2] > dp[e.e1] + e.cost) {
               dp[e.e2] = dp[e.e1] + e.cost;
            }
         }
      }
   }
```



3->4 2
2->3 4
1->2 3