LEC-41

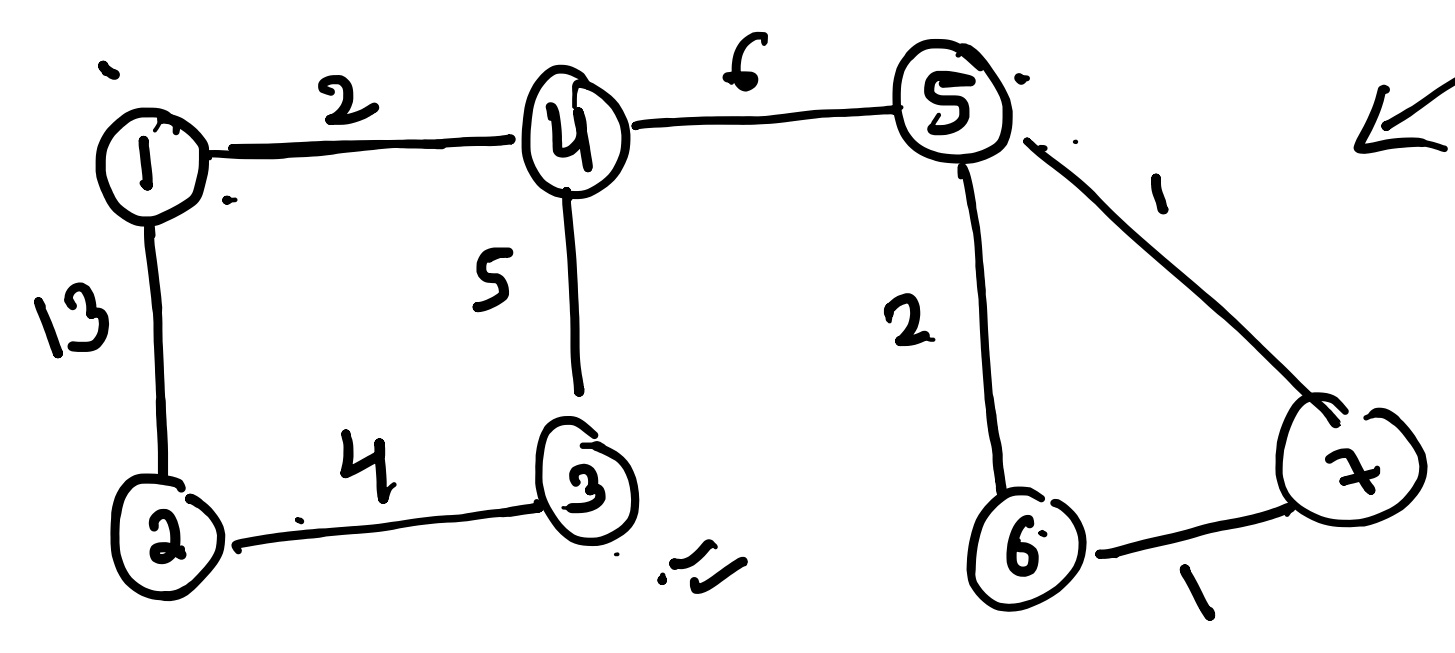
31/05/25 4:57 PM

Single source shortest path

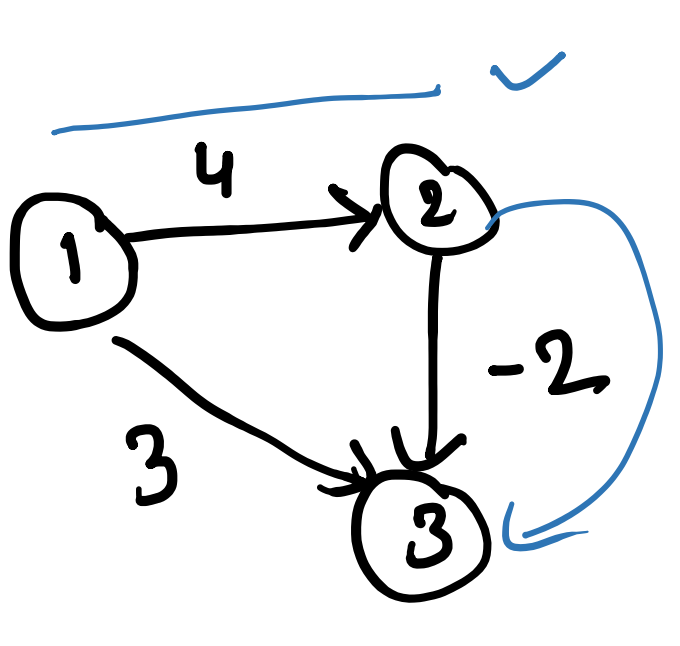
(i) Dijkstra → ✓

(ii) Bellman ford → ✓

1 → 4 ✓
1 → 2 ✓
1 → 7 ✓

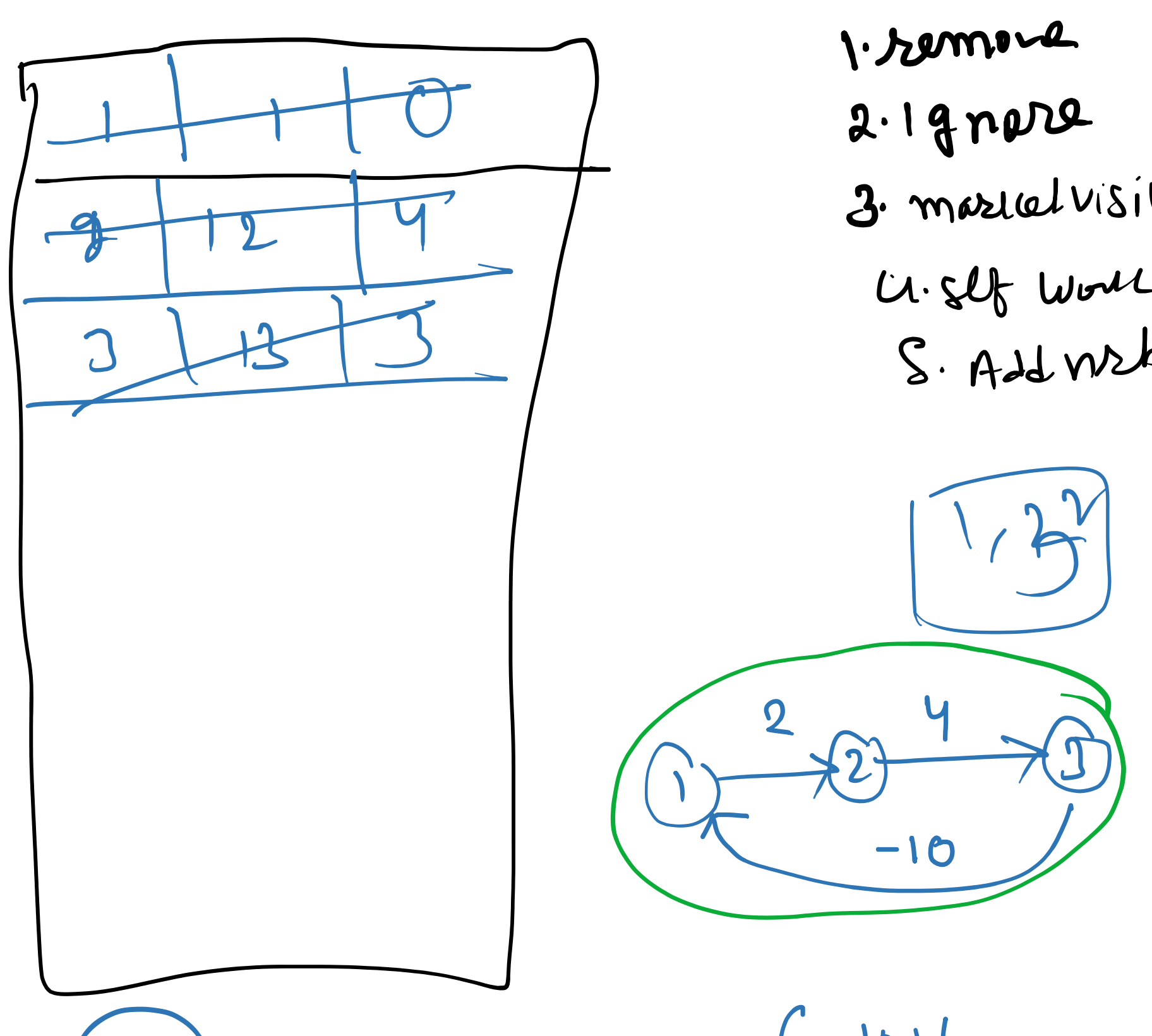


1. remove
2. Ignore
3. marked visited
4. self work
5. Add nbrs



1 1 @ 0
3 13 @ 3
2 12 @ 4

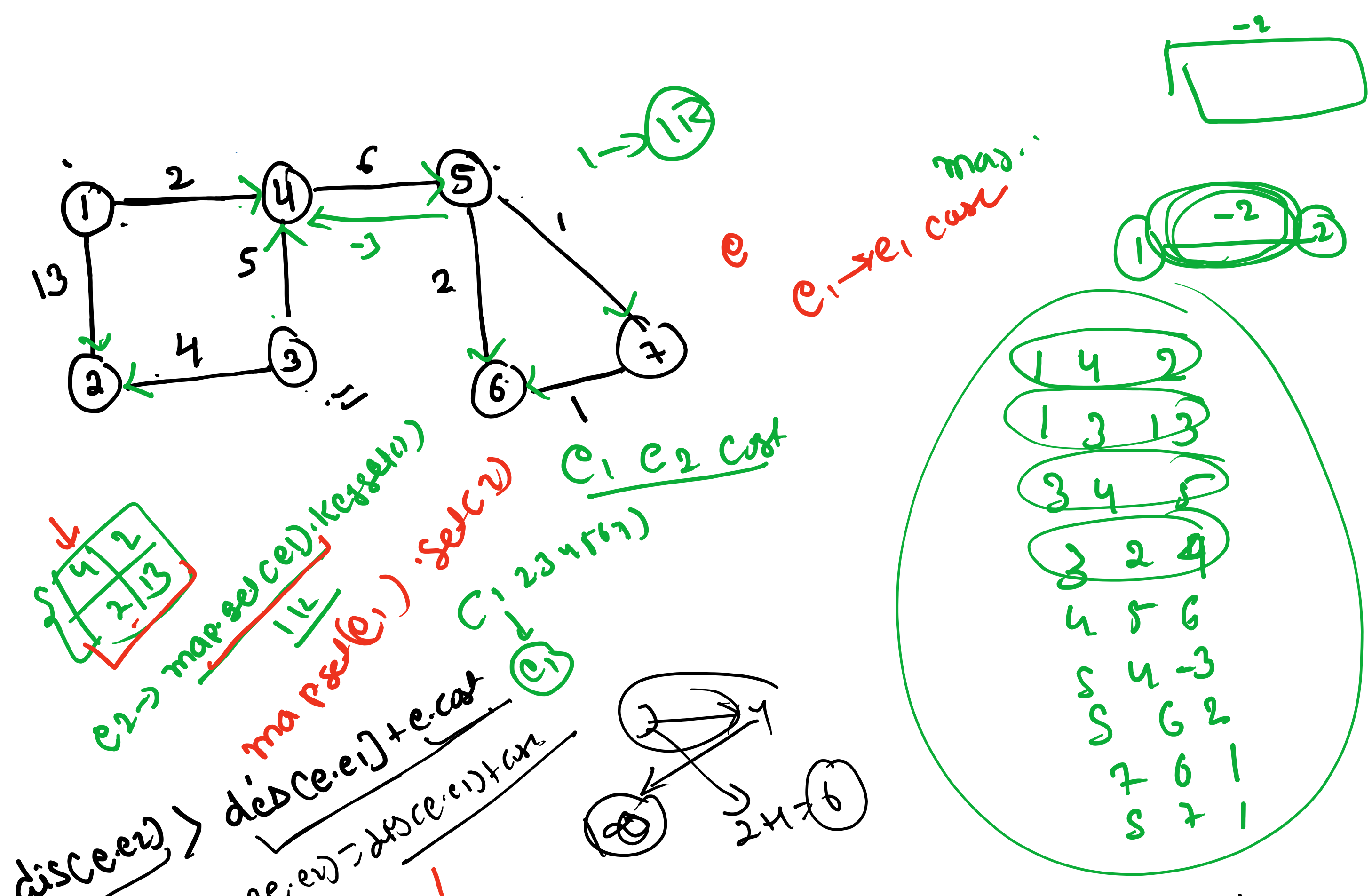
123 → 2



123 → 6
123123 → +9
123123123 → -2

6 − 10 + 6

(i) cycle sum −ve



e₁ → e₁ cost

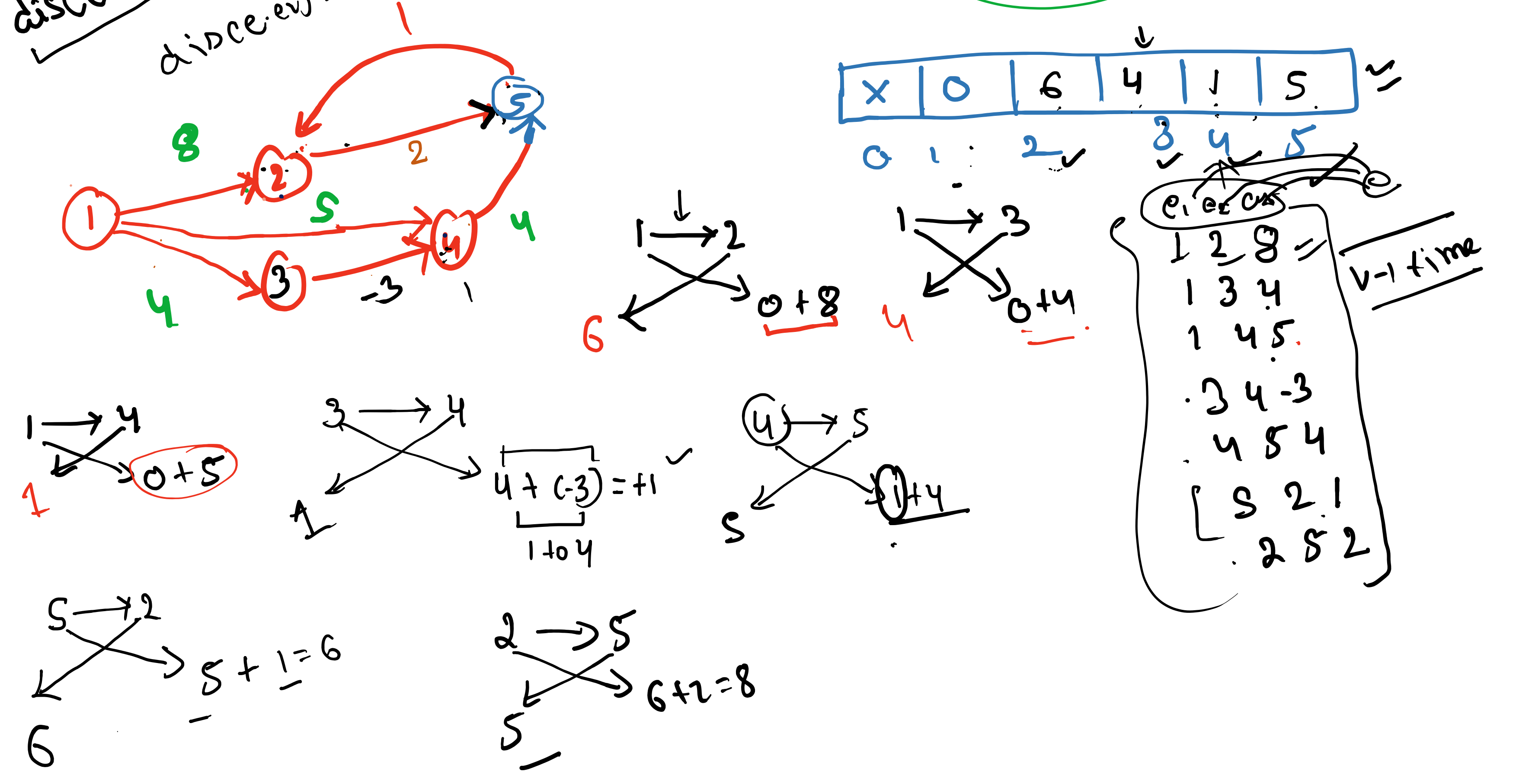
e₂ → map.get(e₁).keySet()

map.get(e₁).get(e₂)

e₁ e₂ cost

dis(e.e2) > dis(e.e1) + e.cost

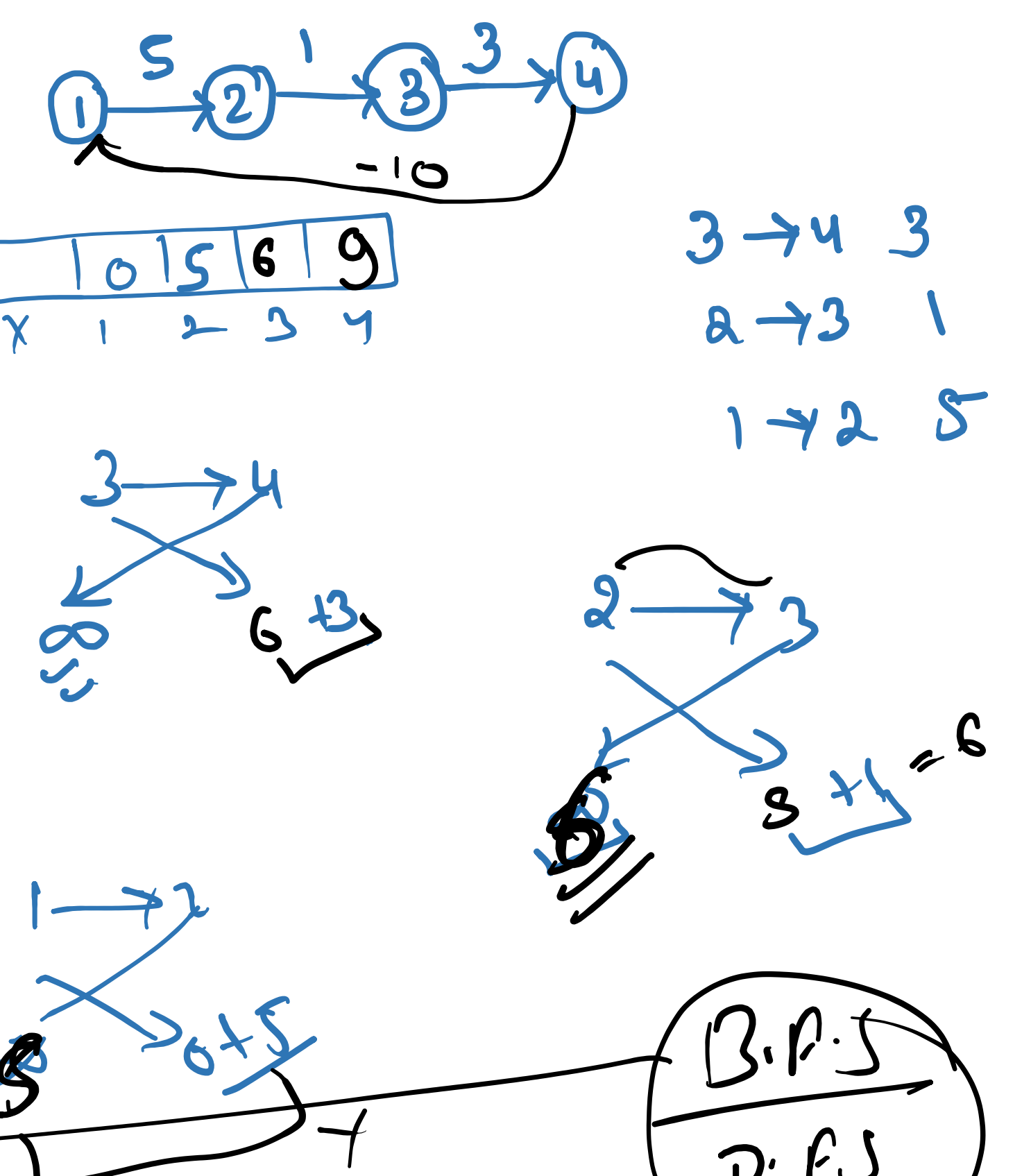
dis(e.e2) = dis(e.e1) + cost



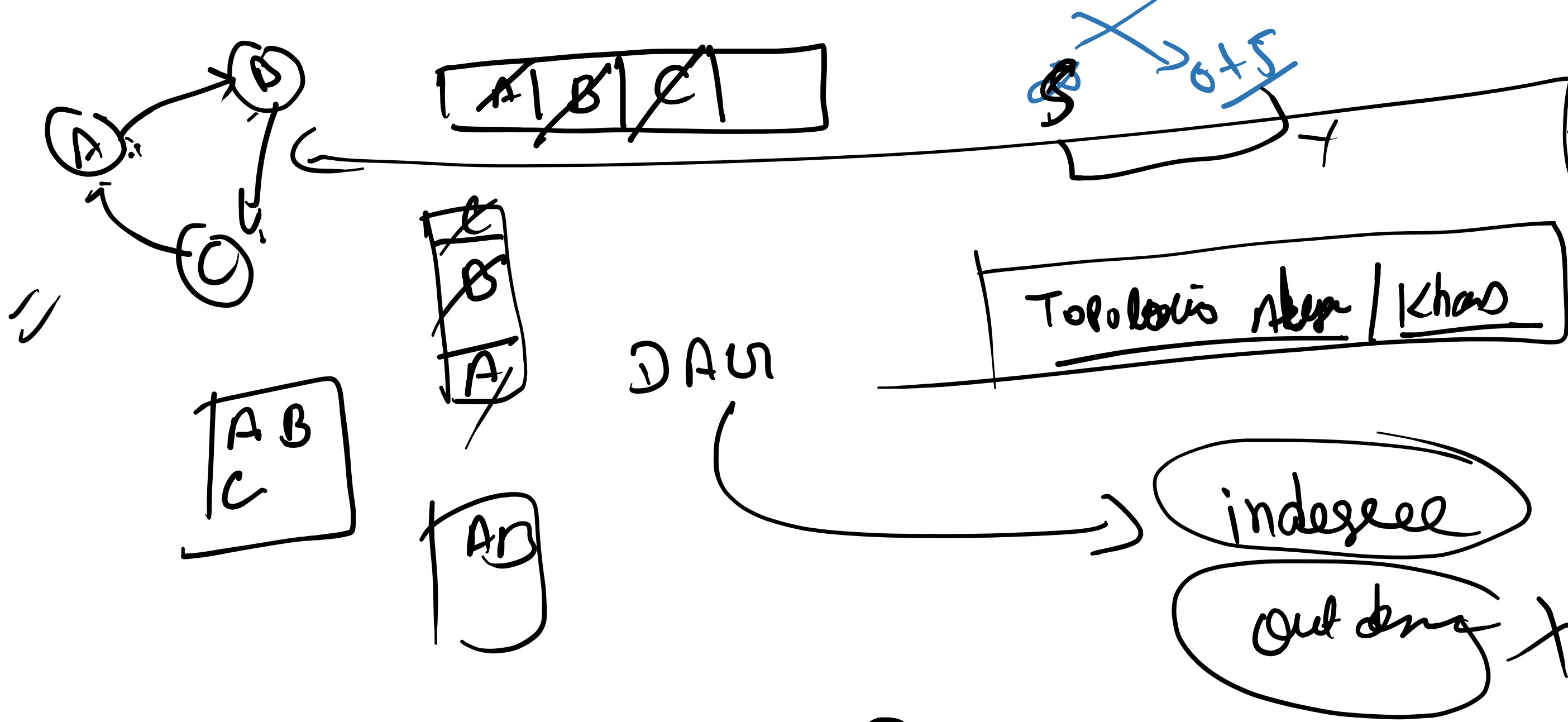
V−1 time

```
public void BellManFord() {
    List<EdgeList> ll = getallEdge();
    int v = map.size();
    int[] dis = new int[v + 1];
    for (int i = 2; i < dis.length; i++) {
        dis[i] = 9999999;
    }
    for (int i = 1; i < v; i++) {
        for (EdgeList e : ll) {
            if (dis[e.e2] > dis[e.e1] + e.cost) {
                dis[e.e2] = dis[e.e1] + e.cost;
            }
        }
    }

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



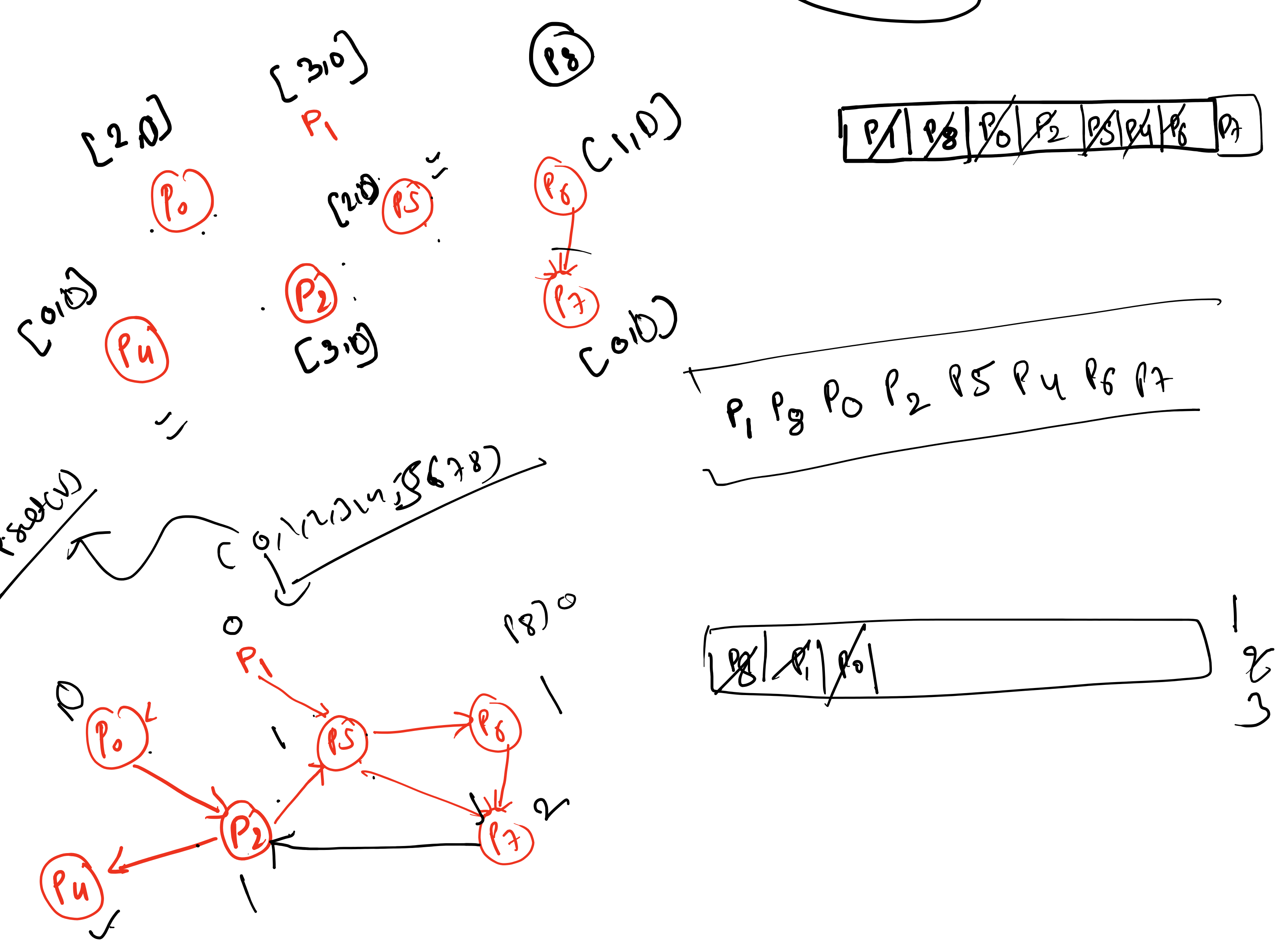
3 → 4 3
2 → 3 1
1 → 2 5



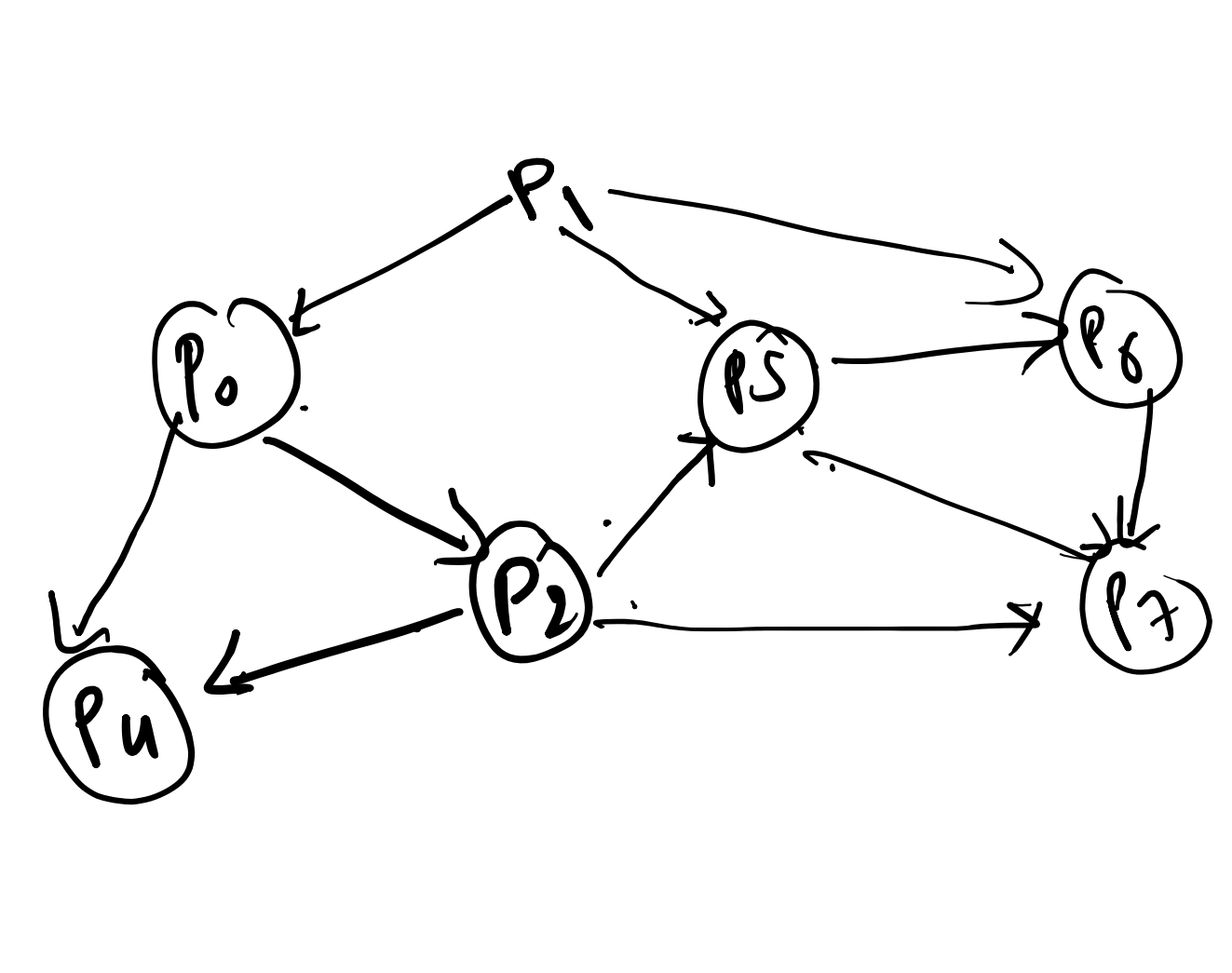
B.F.S / D.F.S

Topological sort | Khan

DAG → indegree / outdegree



P1 P8 P0 P2 P5 P4 P6 P7



Key

Counting

```
int[] arr = { 2, 1, 1, 0, 1, 2, 5, 4, 0, 2, 8, 7, 9, 2, 6, 1, 9};
```

freq(i) = freq(i) + freq(i−1)