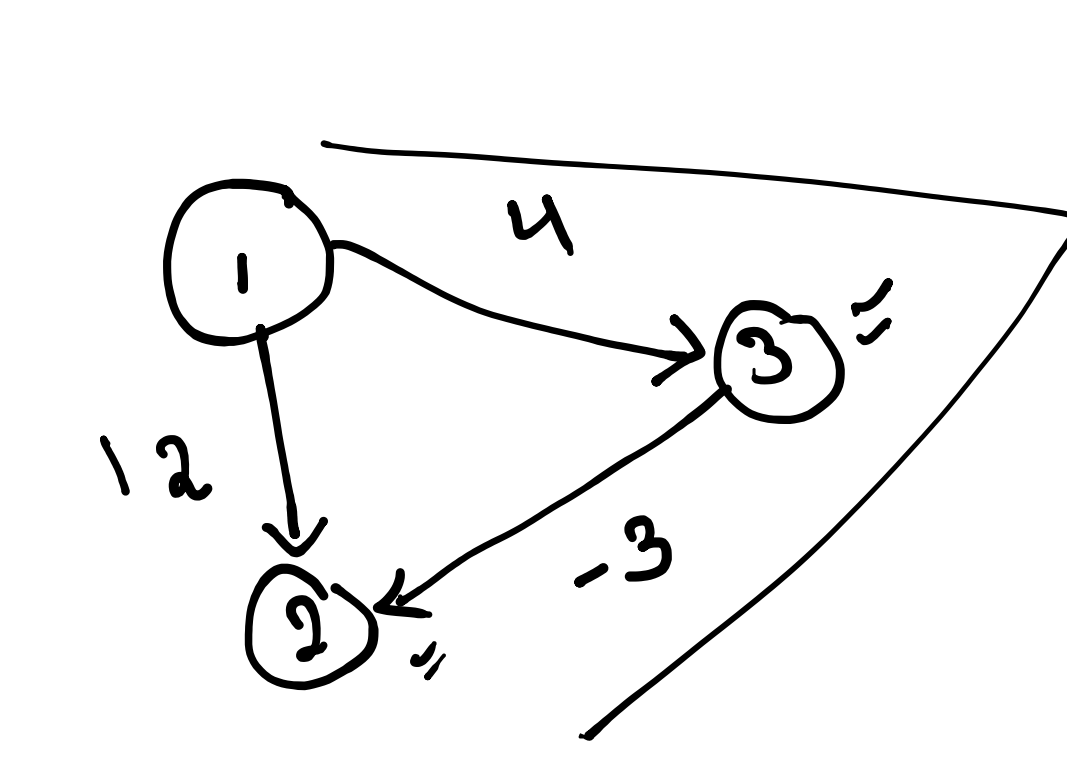


Handwritten list of numbers with circles around them:

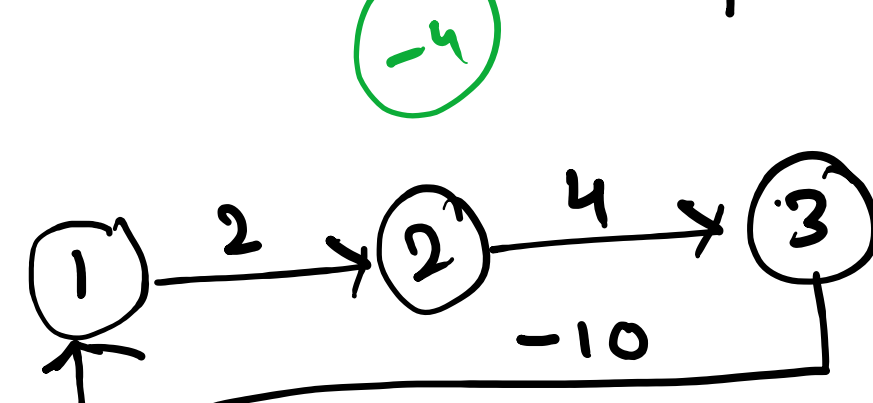
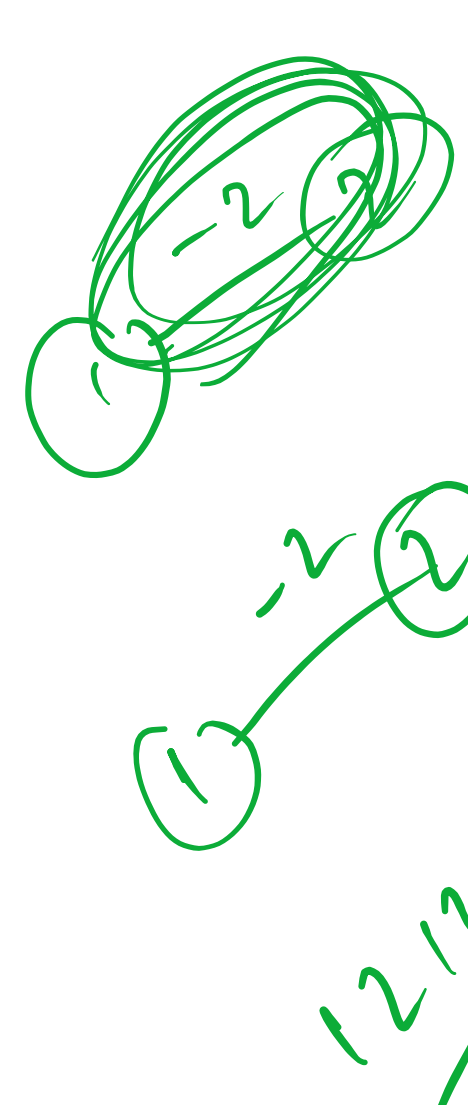
- 1 1 @ 0
- 2 12 @ 2
- 3 17 @ 4
- 2 132 @ 1



$$\begin{array}{r} 3333 \\ \underline{4} \end{array}$$

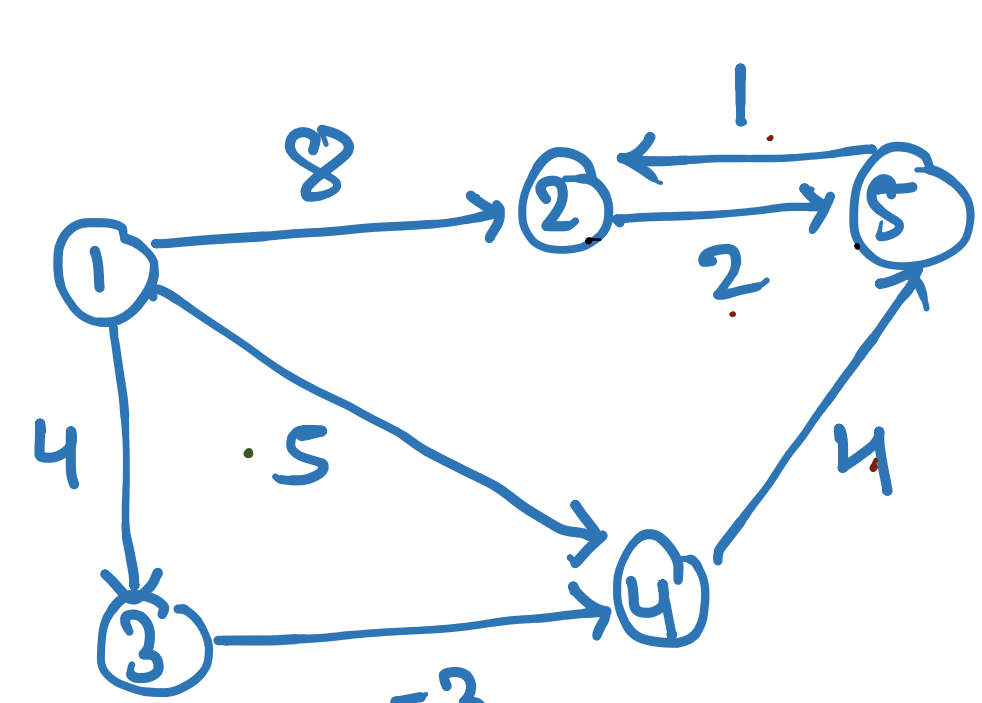
1	1	1	0
2	12	2	
3	13	4	

1,2 ✓
1,3 ✓

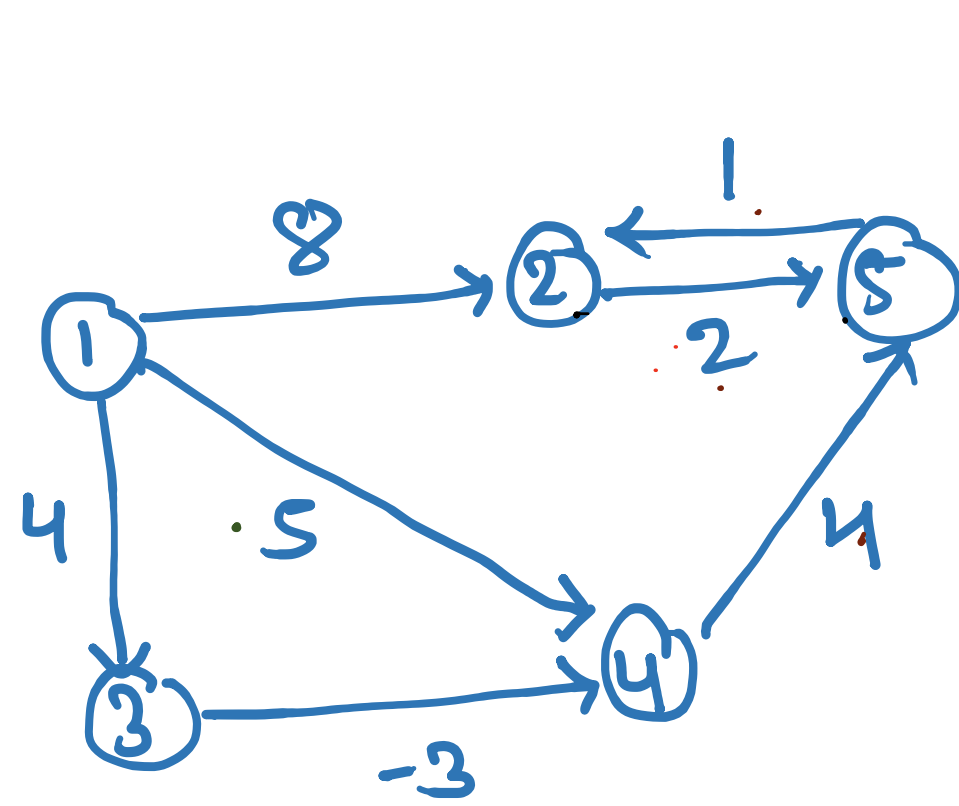


$$\begin{array}{r} 12 \rightarrow 2 \\ 123 \rightarrow 6 \\ \hline 12312 = (-2) \\ 24-102 \\ \hline 12312 \overline{) 312} \rightarrow 6 \\ -24 \quad 4-102 \end{array}$$

$(1, 3, 4), [1, 4, 5], \underline{1, 2, 8}, 3, 4, -3, 2, 5, 2, 5, 2, 1$



$4(5, 4)$
 $\left\{ \begin{array}{l} 1 \rightarrow 3 \ 4 \\ 1 \rightarrow 2 \ 8 \\ 1 \rightarrow 4 \ 5 \end{array} \right.$
 $func e1: map.keySet() \&$
 $func e2: map.get(e1).keySet()$
 $csd = \underline{map.get(e1).get(e2)}$
 $\left\{ \begin{array}{l} 3 \rightarrow 4 \ -3 \\ 4 \rightarrow 5 \ 4 \\ 2 \rightarrow 5 \ 5 \\ 5 \rightarrow 2 \ 1 \end{array} \right.$

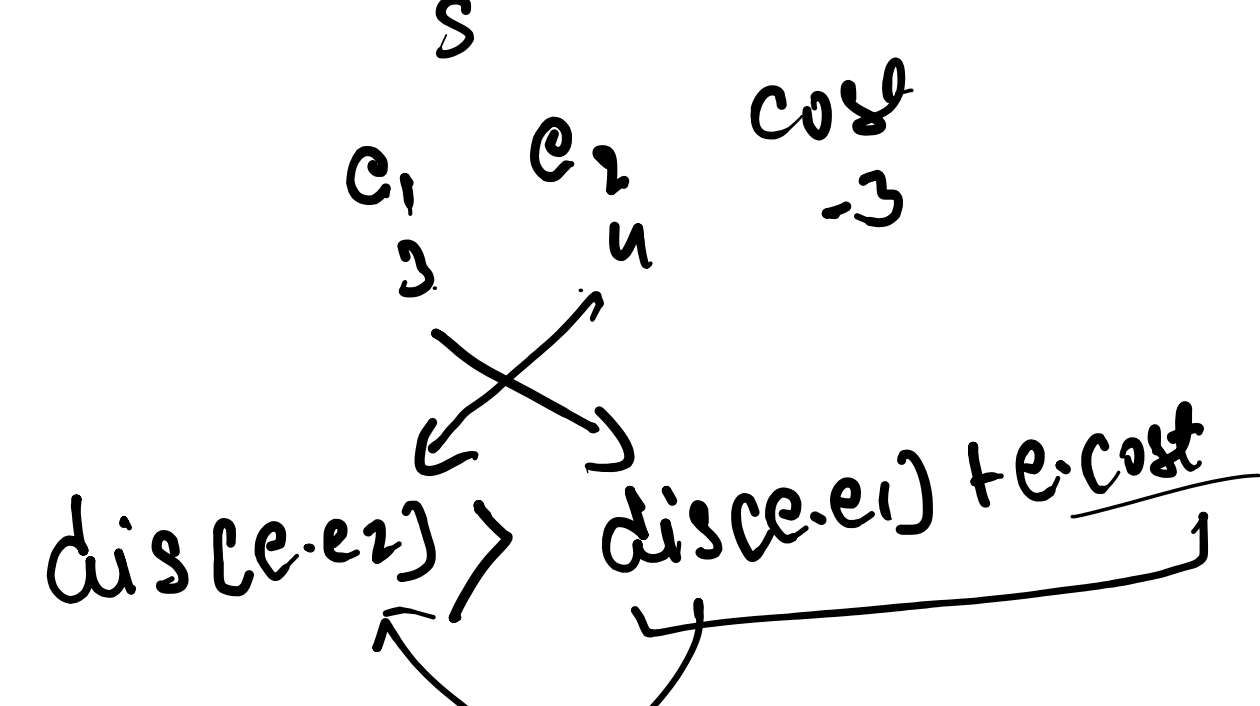
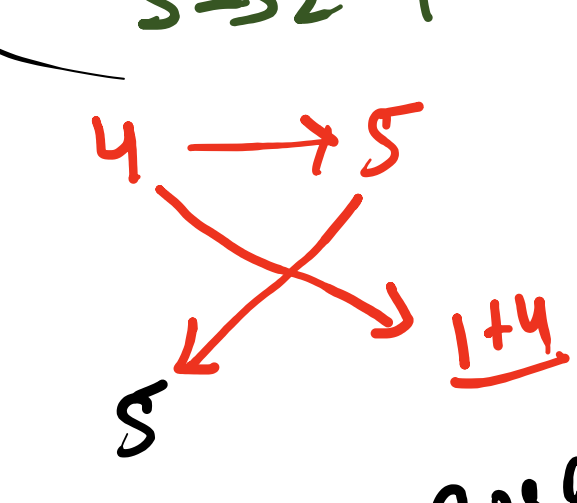
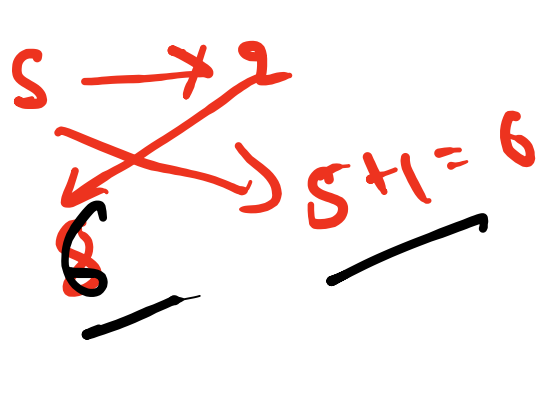
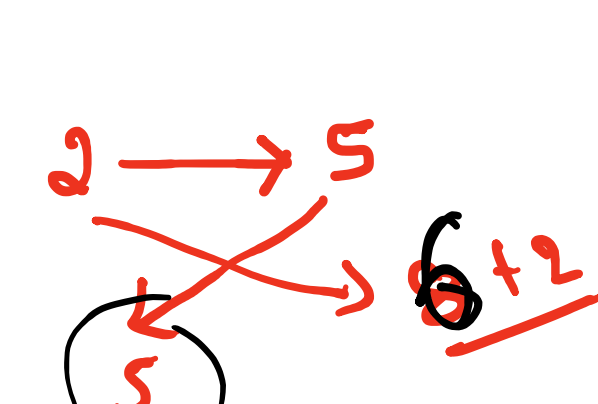
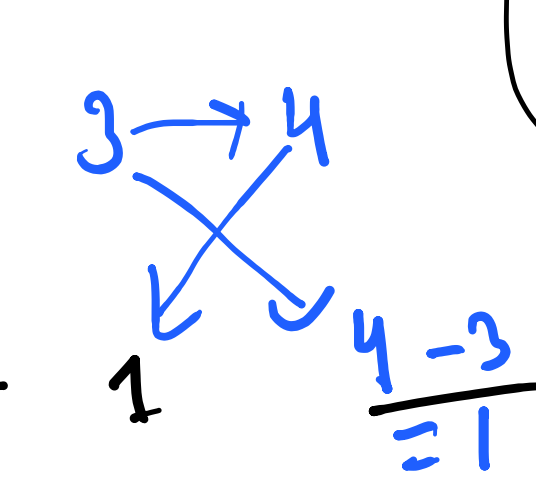
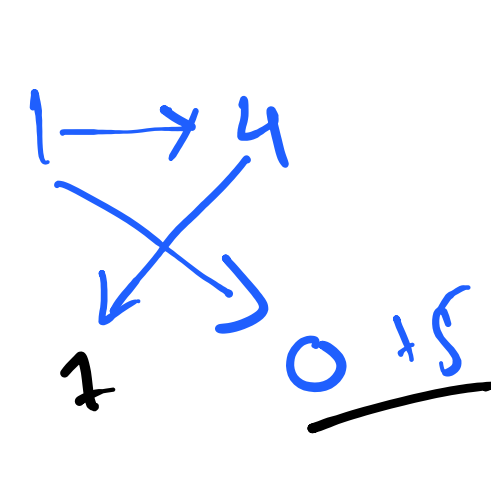
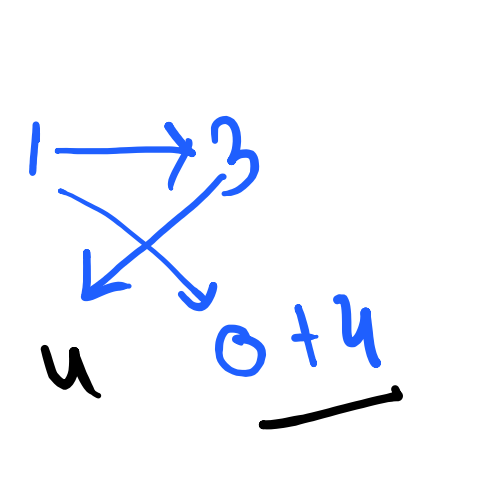
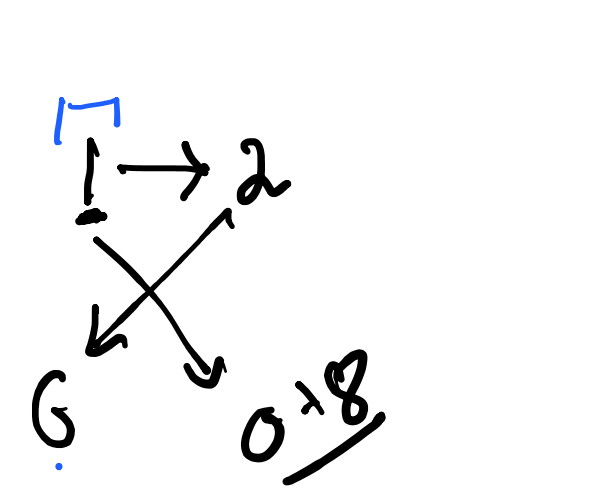


↓

	0	6	4	1	5
--	---	---	---	---	---

$1 \rightarrow 2 \quad 8 \quad \checkmark$
 $1 \rightarrow 3 \quad 4$
 $1 \rightarrow 4 \quad 5$
 $3 \rightarrow 4 \quad 3$
 $4 \rightarrow 5 \quad 4$
 $2 \rightarrow 5 \quad 2$
 $5 \rightarrow 2 \quad 1$

?
 $\frac{v-1}{2}$ time
RIK

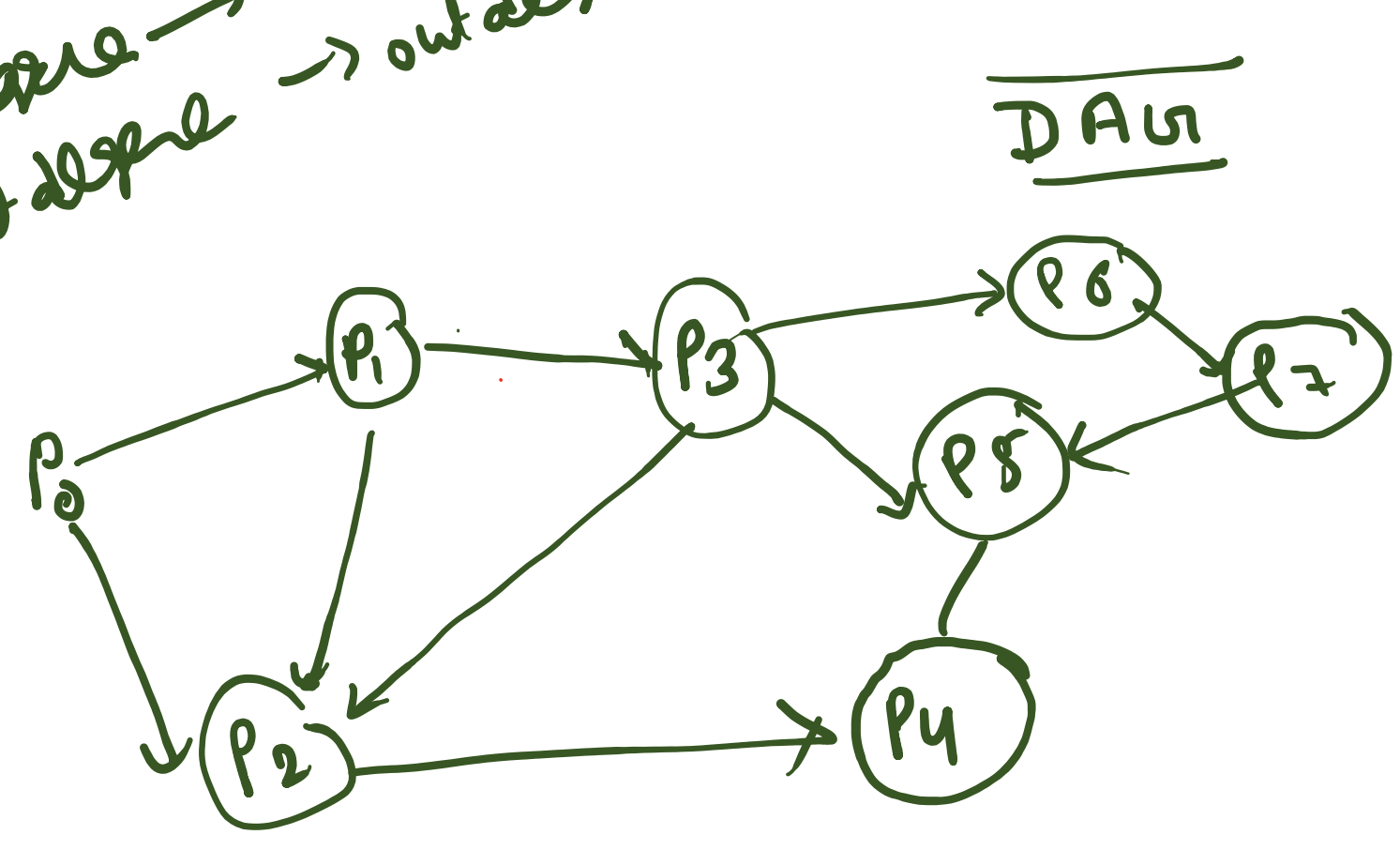


```

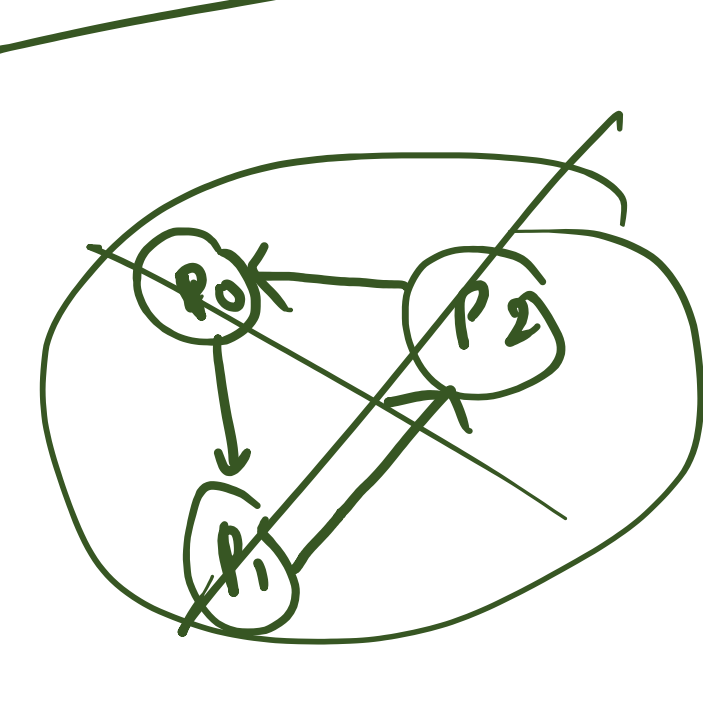
public void BellMan_Ford() {
    List<EdgePair> ll = getALLEdge();
    int v = map.size();
    int[] dis = new int[dis.length];
    for (int i = 2; i < dis.length; i++) {
        dis[i] = 999999;
    }
    for (int i = 1; i < v; i++) {
        for (EdgePair e : ll) {
            if (dis[e.e2] > dis[e.e1] + e.cost)
                dis[e.e2] = dis[e.e1] + e.cost;
        }
    }
    for (int i = 1; i < dis.length; i++) {
        System.out.println(i + " " + dis[i]);
    }
}

```

indegree \rightarrow incoming
outdegree \rightarrow outgoing



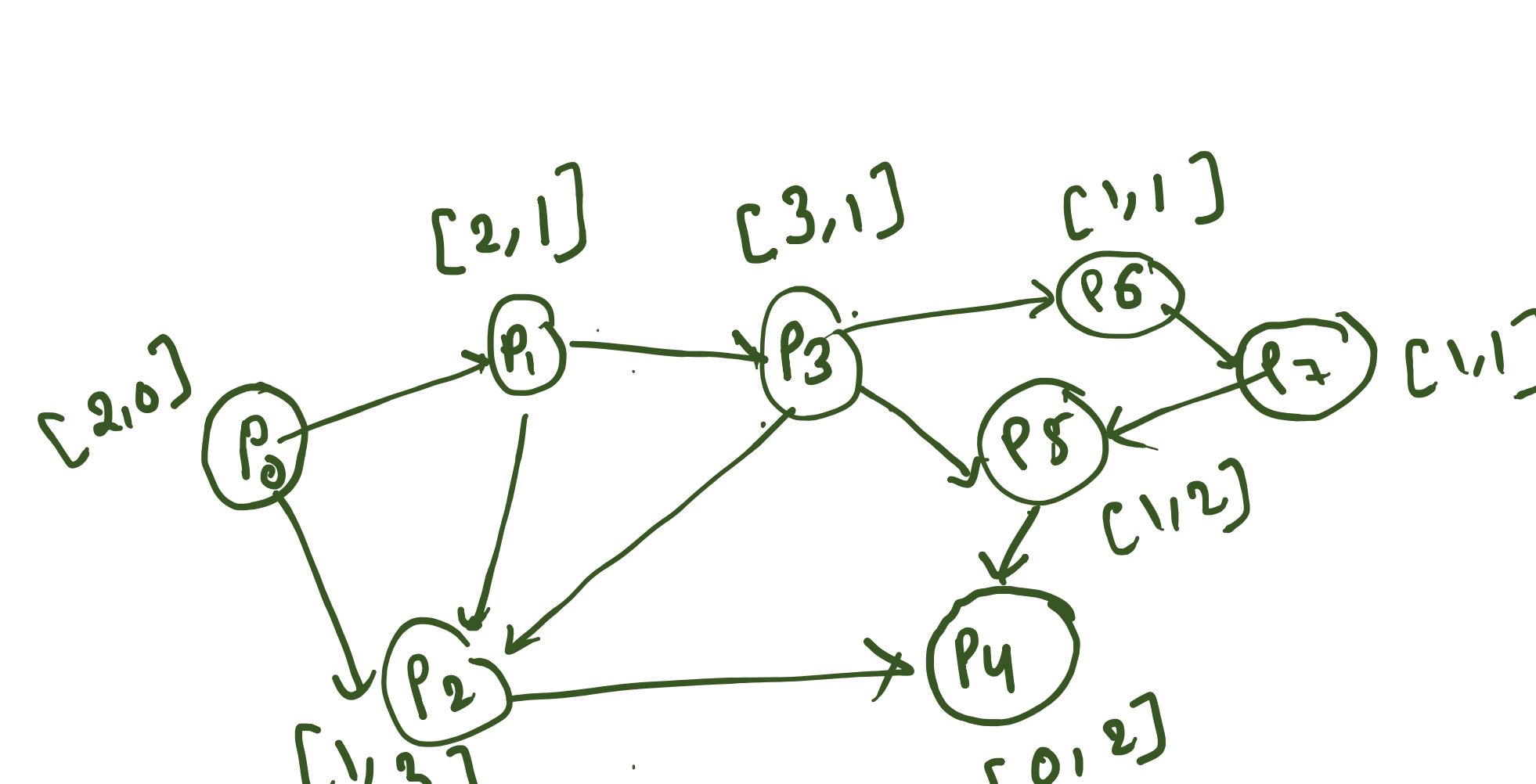
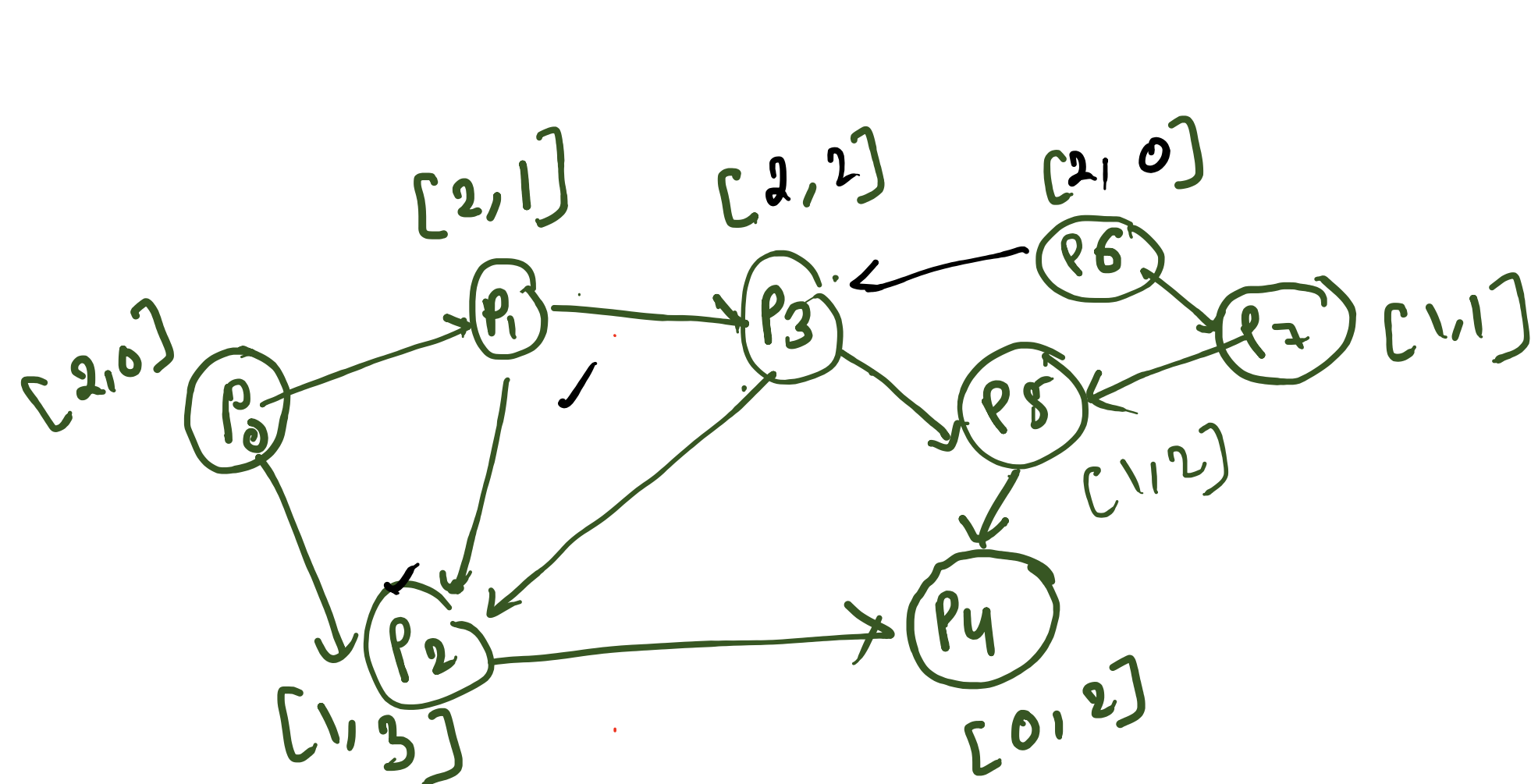
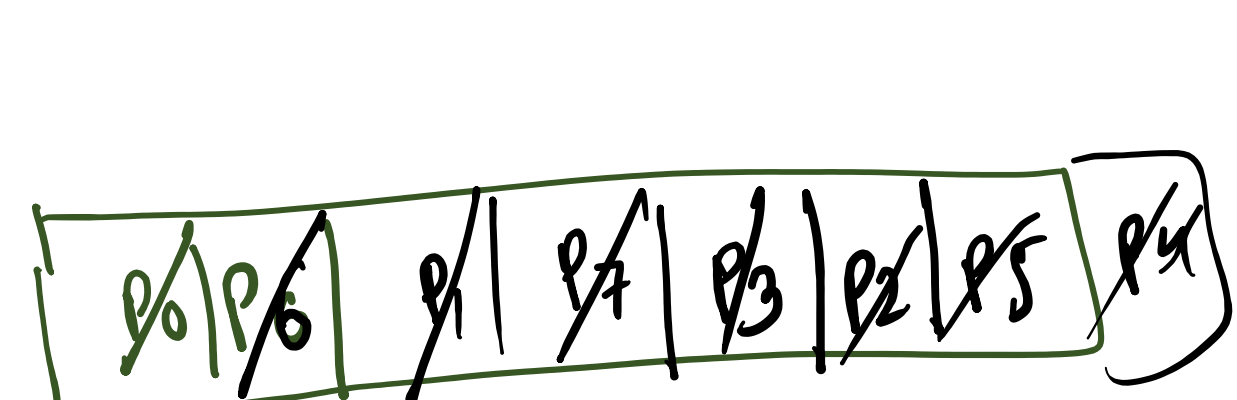
Topological sort
khans algo



P/L P/L P/L

PO P1

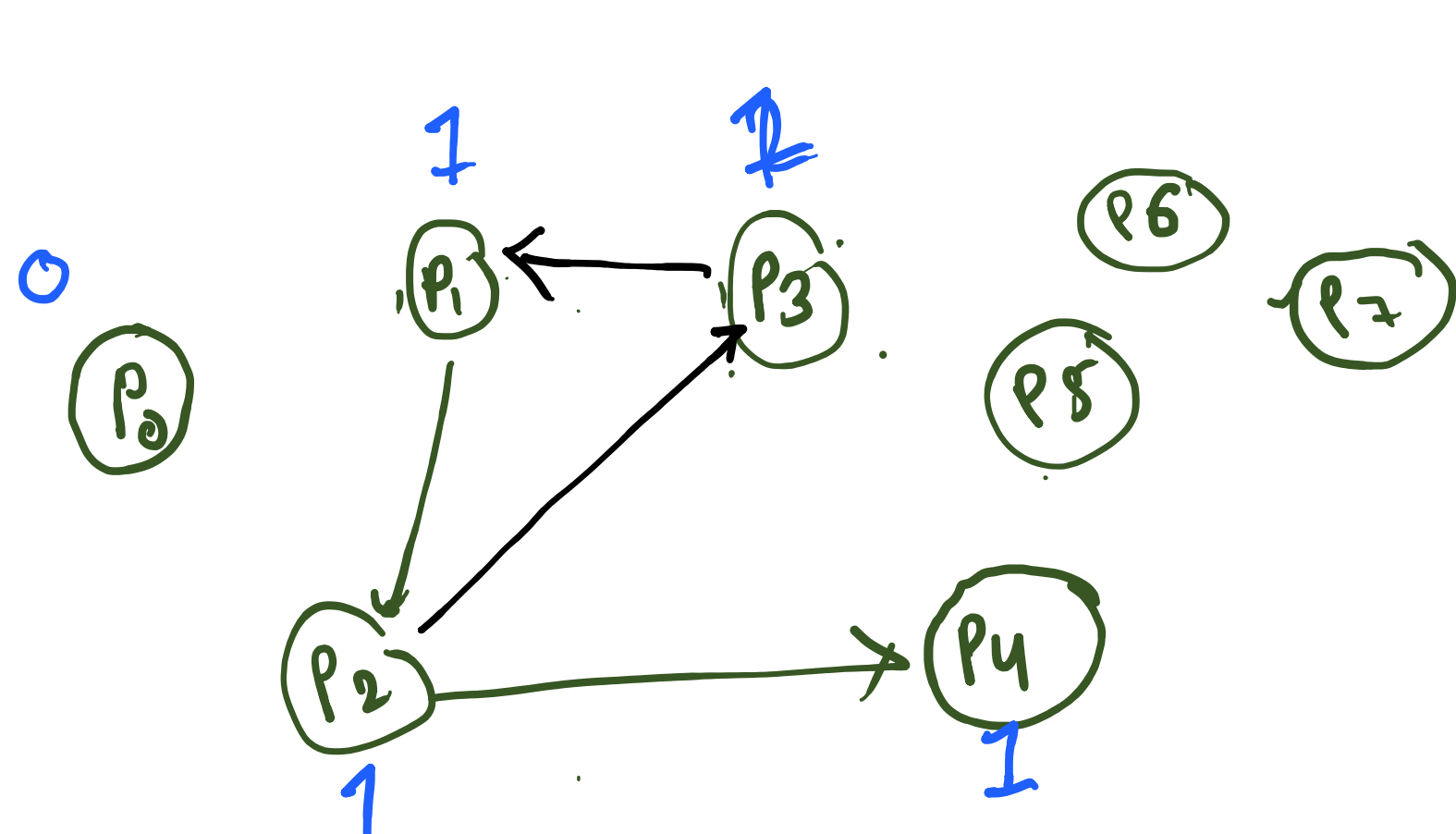
$p_0, p_6, p_1, p_7, p_2, p_3, p_5, p_4$



```

public void TopologicalSort() {
    int[] in = Indegree();
    Queue<Integer> q = new LinkedList<>();
    for (int i = 0; i < in.length; i++) {
        if (in[i] == 0) {
            q.add(i);
        }
    }
    while (!q.isEmpty()) {
        int r = q.poll();
        System.out.print(r + " ");
        for (int nbrs : map.get(r)) {
            in[nbrs]--;
            if (in[nbrs] == 0) {
                q.add(nbrs);
            }
        }
    }
    System.out.println();
}

```



<u>Counting</u>	<u>Cost</u>
-----------------	-------------

key

134

$SS \rightarrow 3.5$
 $SN \rightarrow 6 \checkmark$
 $SSC \rightarrow 1000 \checkmark$

max = 9

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

2	4	4	0	1	1	1	1	1	2
0	1	2	3	4	5	6	7	8	9

→

0	1	2	3	4	5	6	7	8	9
1	2	6	10	101	14	12	13	4	5

0	0	1	1	1	1	2	2	2	2	4	5	6	7	8	9	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---