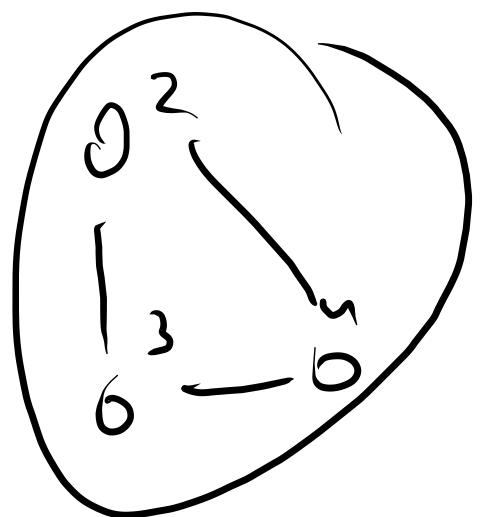
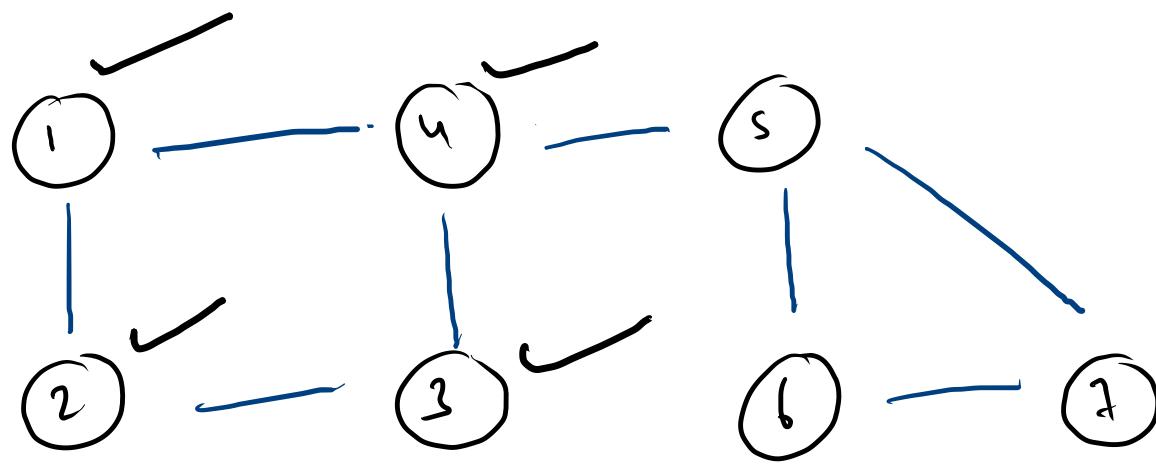
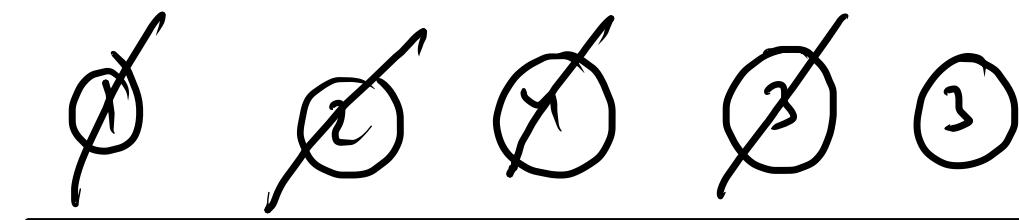


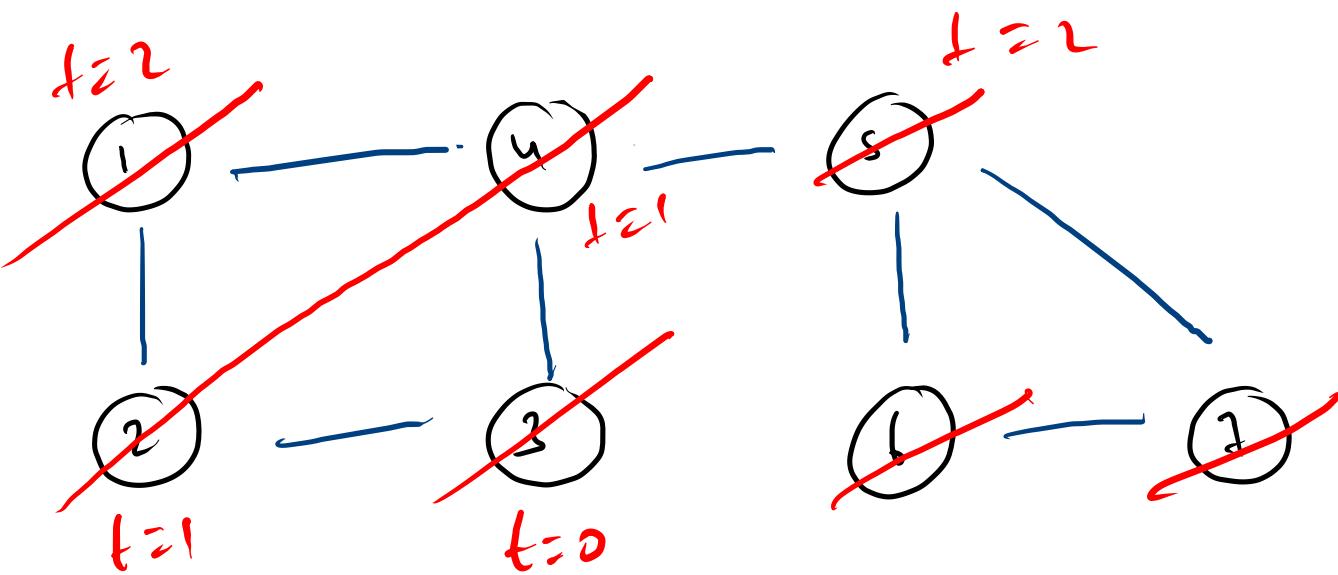
00  
—  
10





8 marks (hill)





$\text{src} \rightarrow 3$

$t=0$

curr

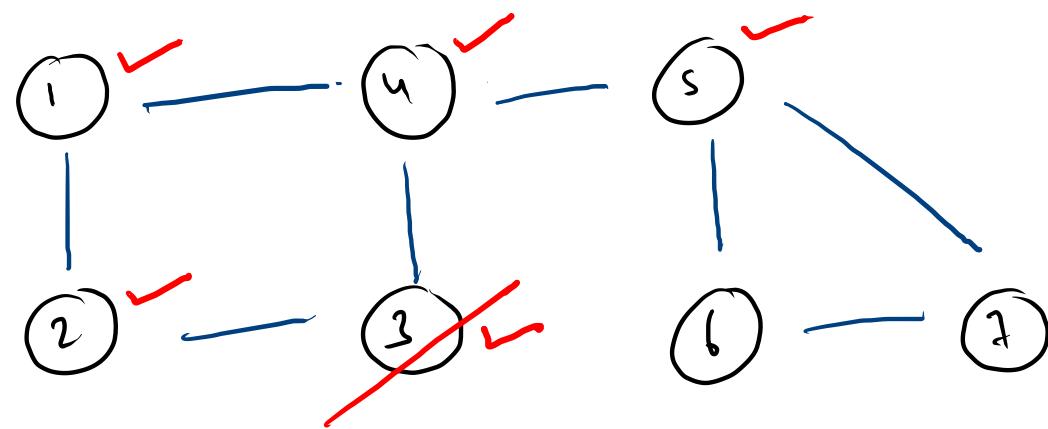
$t=2$

$s$

$t=3$

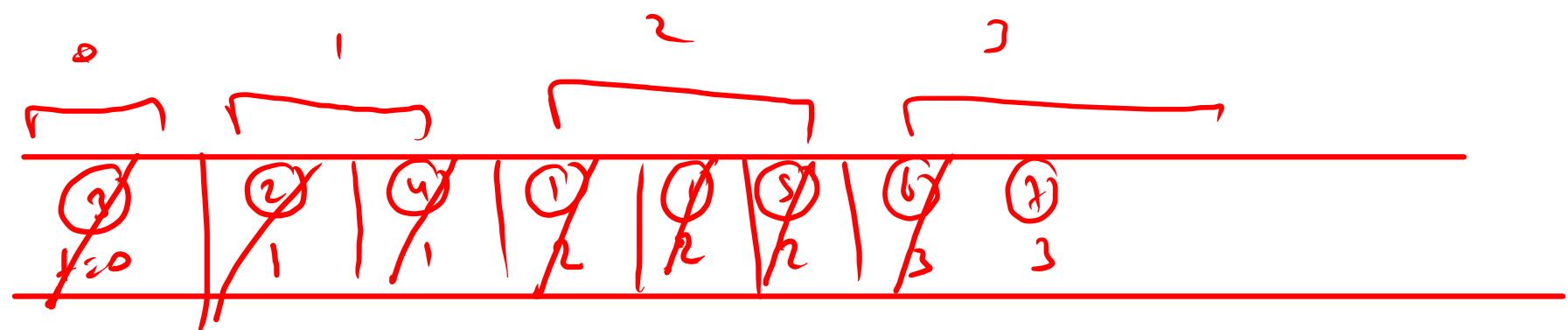
7

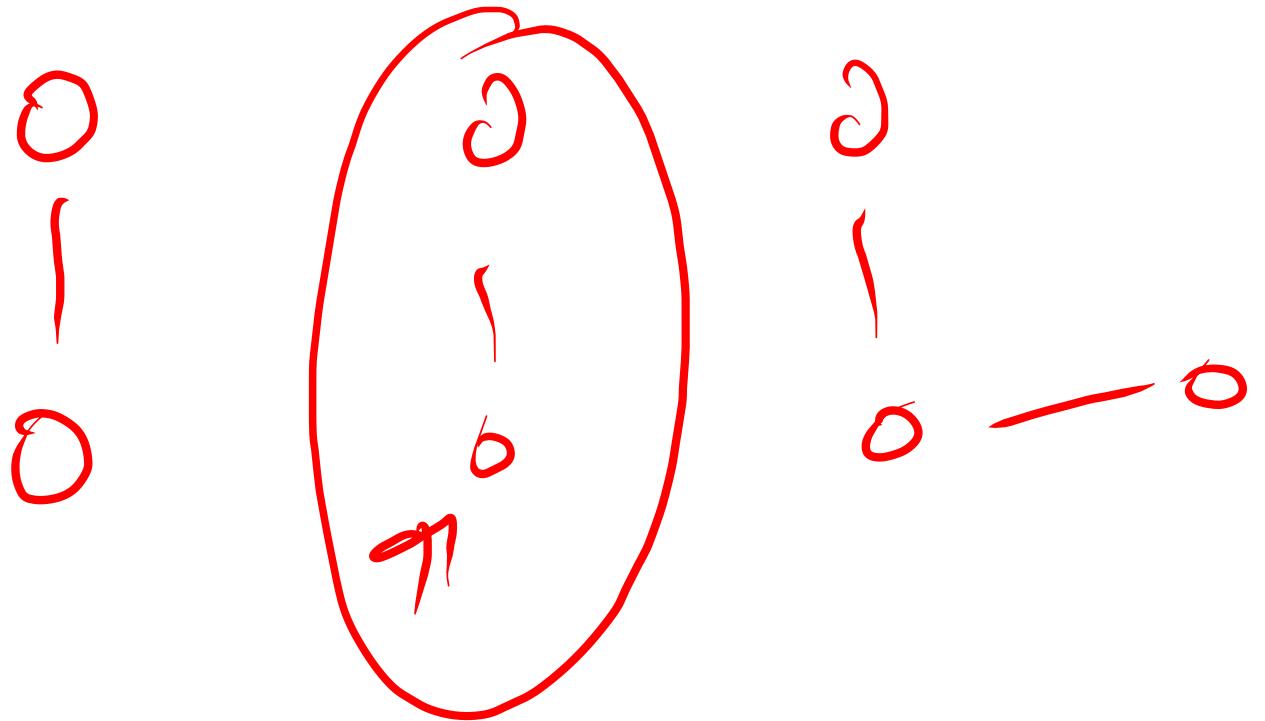
γ mark worle child

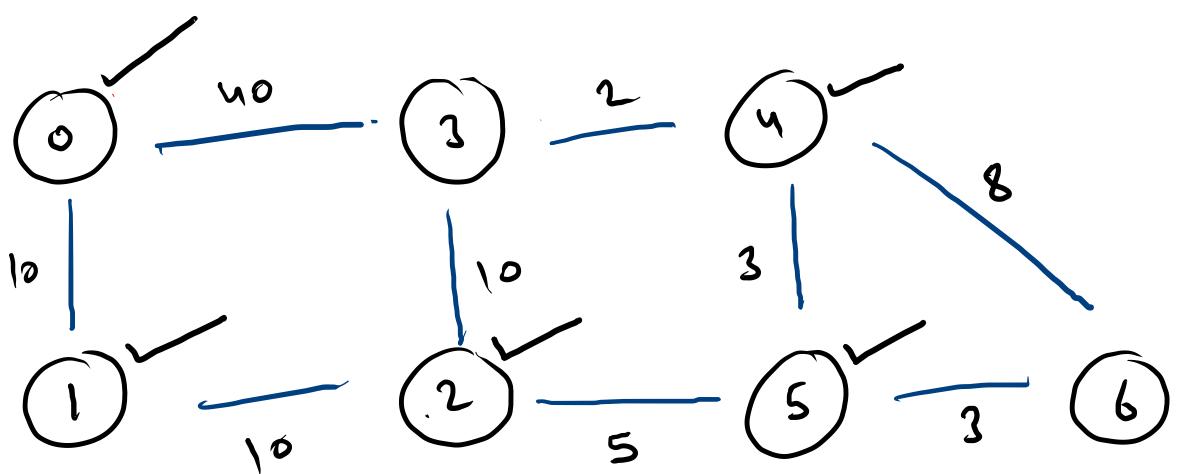


com = 1 2 3 4 5

t=2





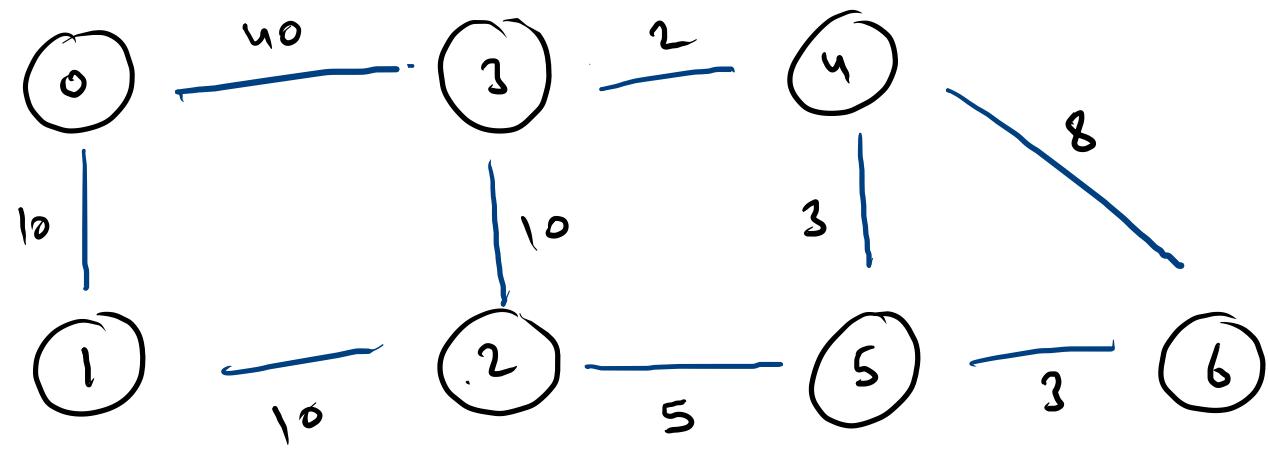


$0 @ 0 \quad 0$   
 $1 @ 10 \quad 01$   
 $2 @ 20 \quad 012$   
 $3 @ 30 \quad 0123$

$s \text{ via } 0125 @ 25$   
 $u \text{ via } 01254 @ 28$

γ mane won't chile

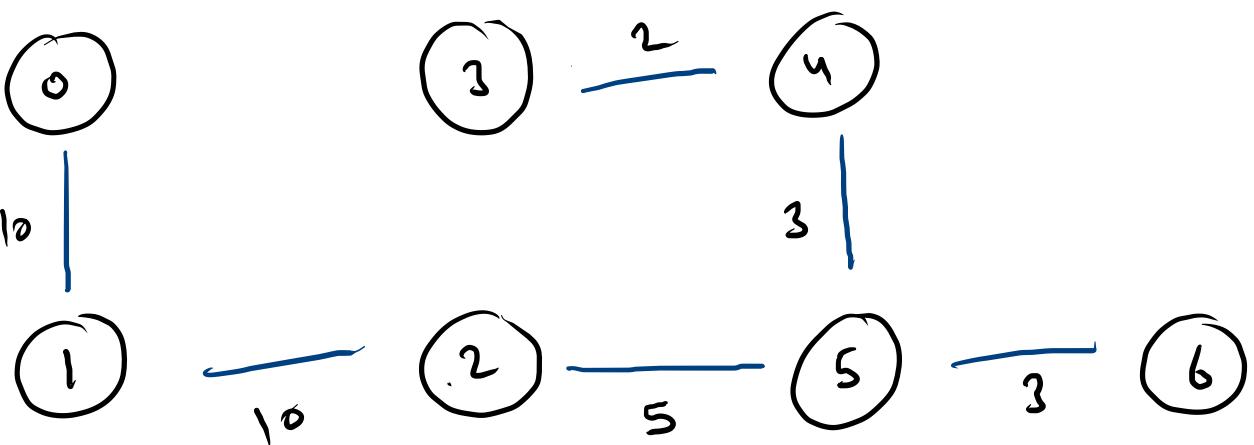
~~u / 25 / "01254"~~   ~~s / 25 / "0125"~~  
 $3 / 40 / "03"$     $6 / 31 / "012546"$   
 $6 / 28 / "01256"$   
 $3 / 30 / "012543"$   
 $3 / 30 / "0123"$

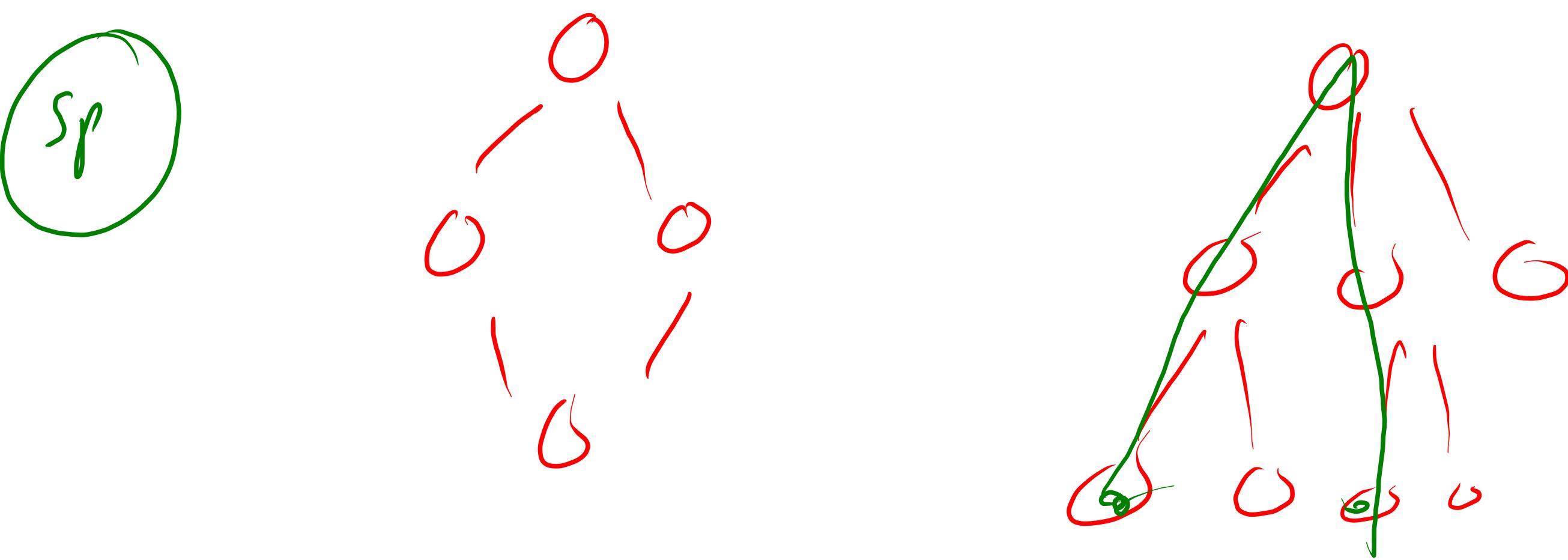


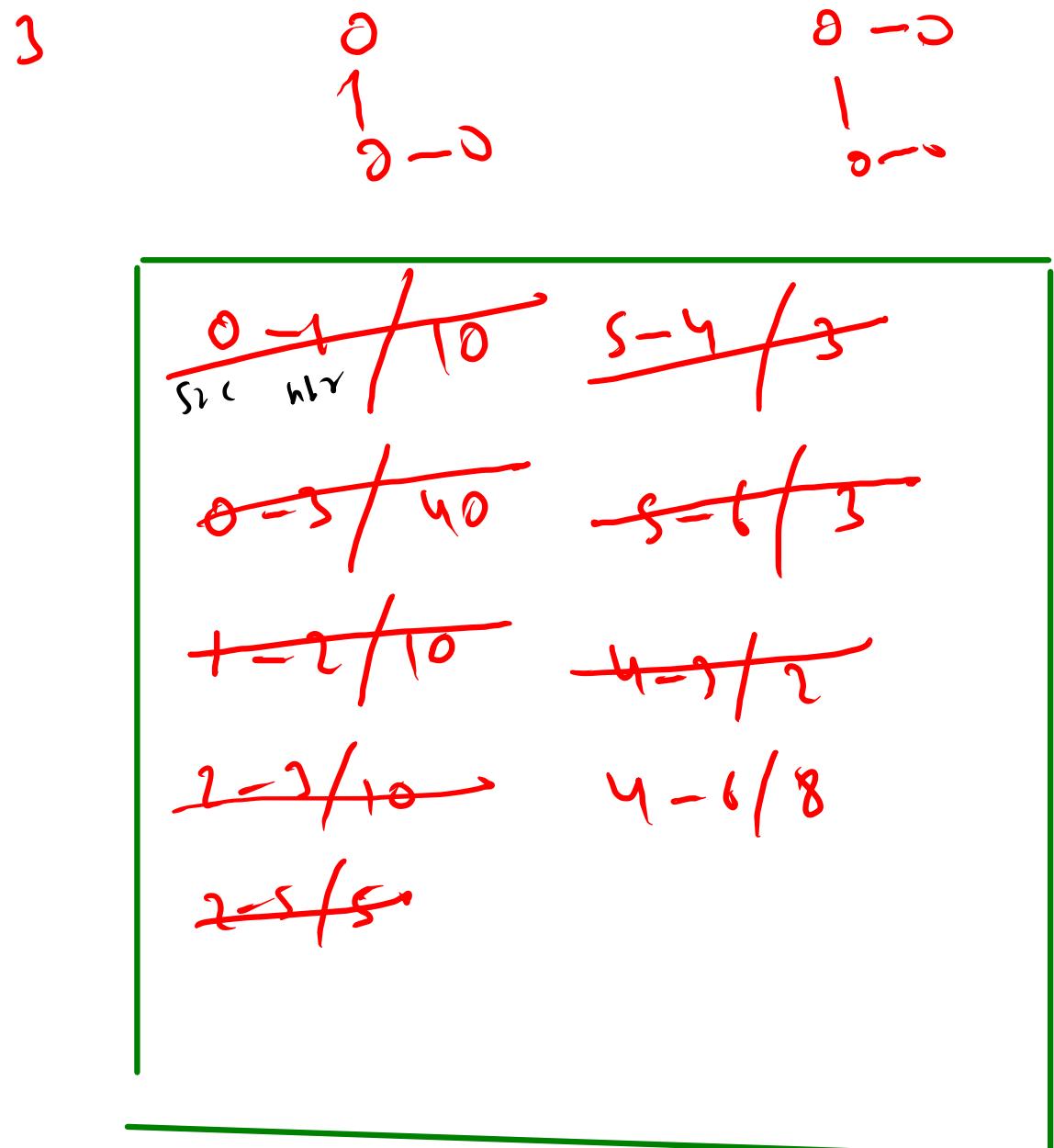
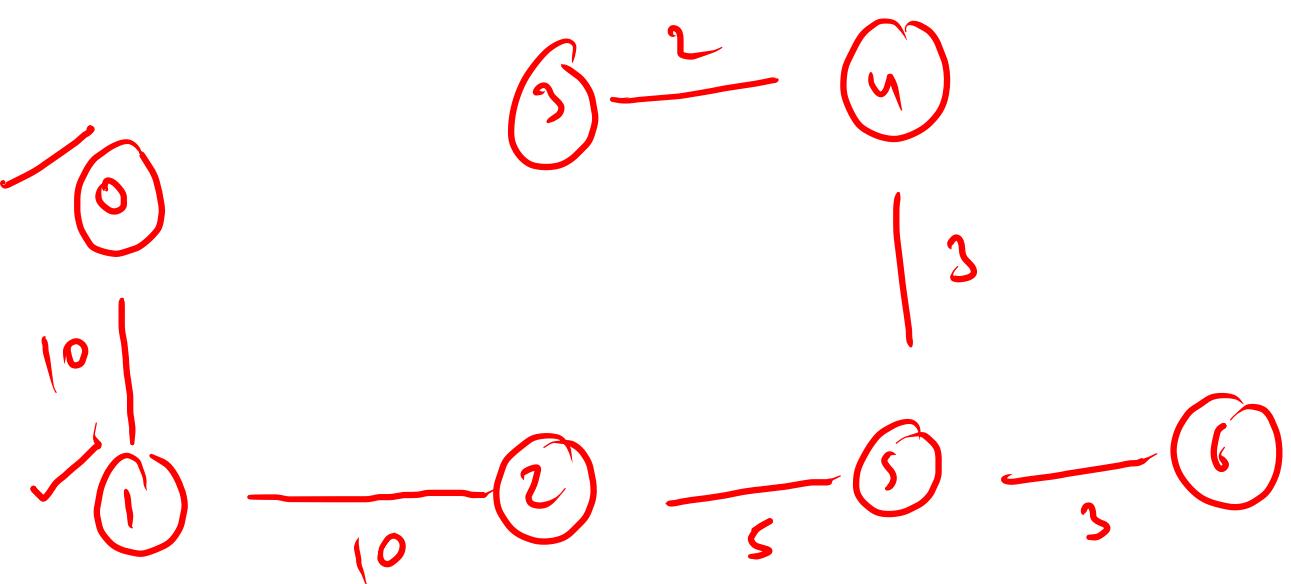
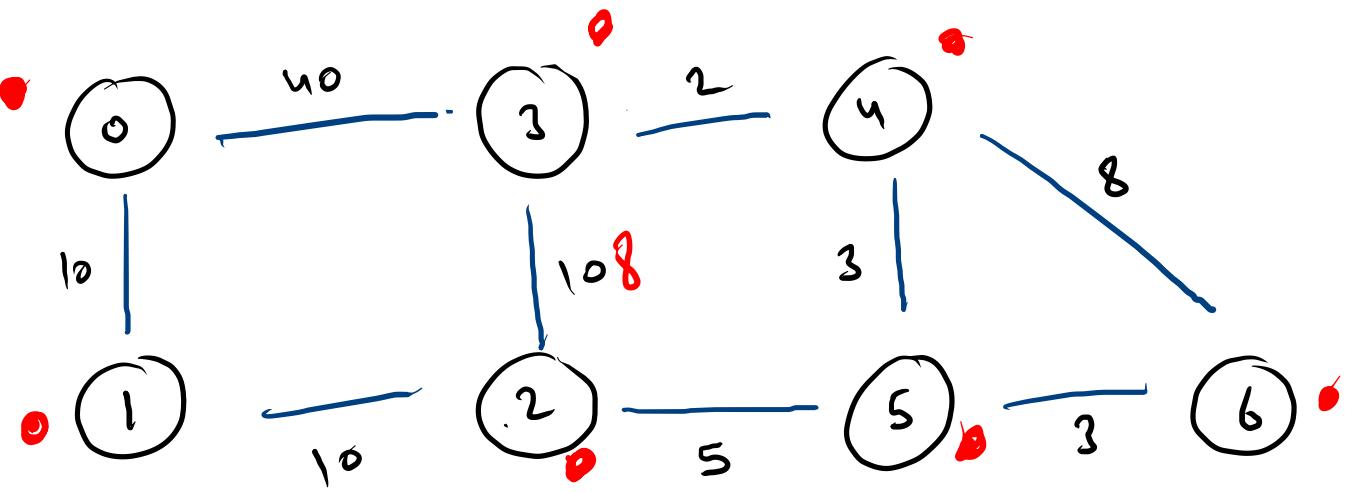
MST

ST

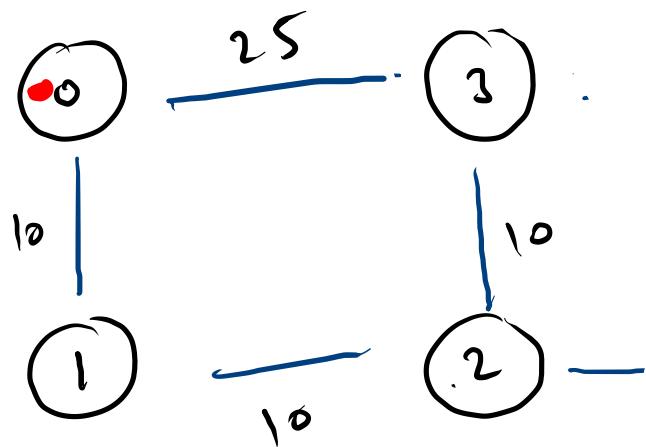
no cycle



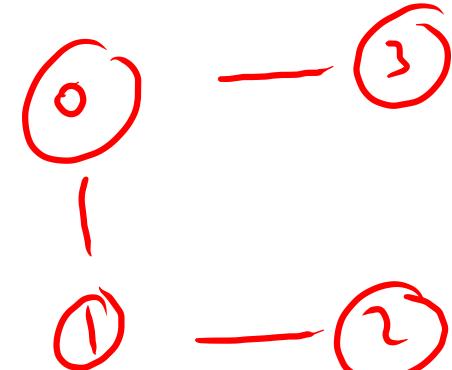






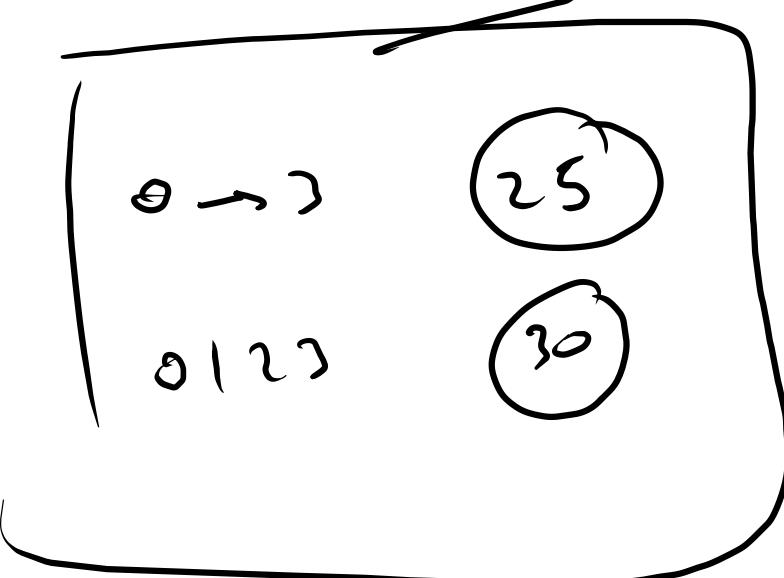


Dikes

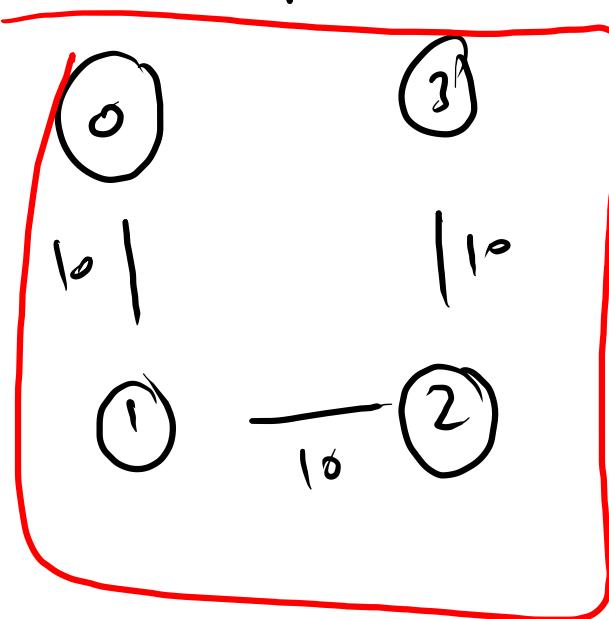


path  
Delhi min  
can to  
all dikes

path Dikes



train



Connectivity

(Ex)  
railways all over  
India

```

PriorityQueue<Pair> pq = new PriorityQueue<>();
pq.add(new Pair(0, 0, 0));

boolean visited[] = new boolean[vtes];
while(pq.size() > 0){

    Pair p = pq.remove();

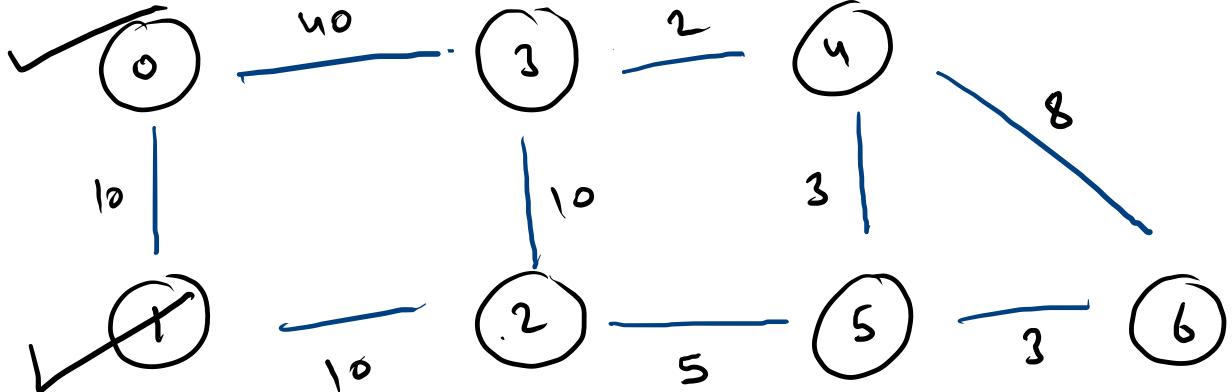
    if(visited[p.nbr]){
        continue;
    }

    visited[p.nbr] = true;

    if(p.src != p.nbr)
        System.out.println("[ "+p.nbr+" - "+p.src+" @ "+p.wt+"]");

    for(Edge ed: graph[p.nbr]){
        if(visited[ed.nbr] == false){
            pq.add(new Pair(p.nbr, ed.nbr, ed.wt));
        }
    }
}

```



1 - 0 @ 10

```

static class Pair implements Comparable<Pair> {
    int src;
    int nbr;
    int wt;

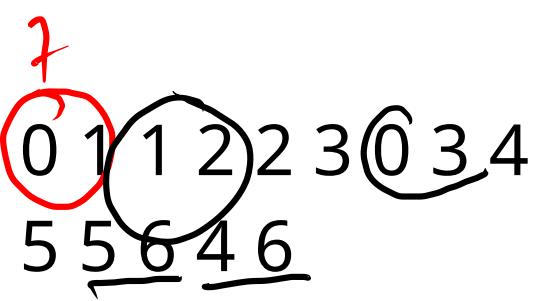
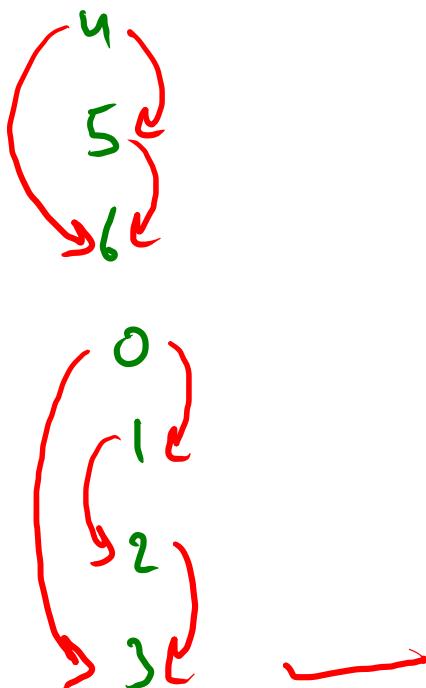
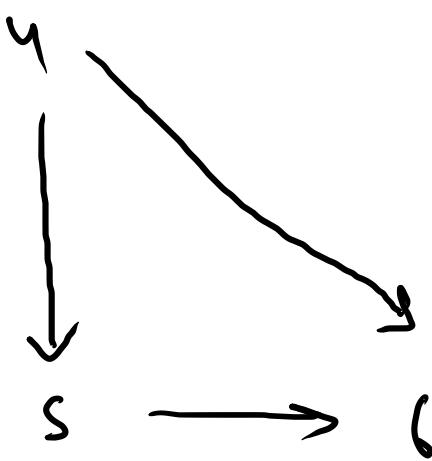
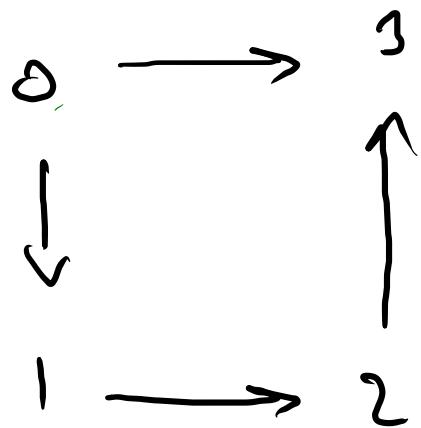
    Pair(int v, int u, int w){
        this.src = v;
        nbr = u;
        wt = w;
    }

    public int compareTo(Pair b){
        return this.wt - b.wt;
    }
}

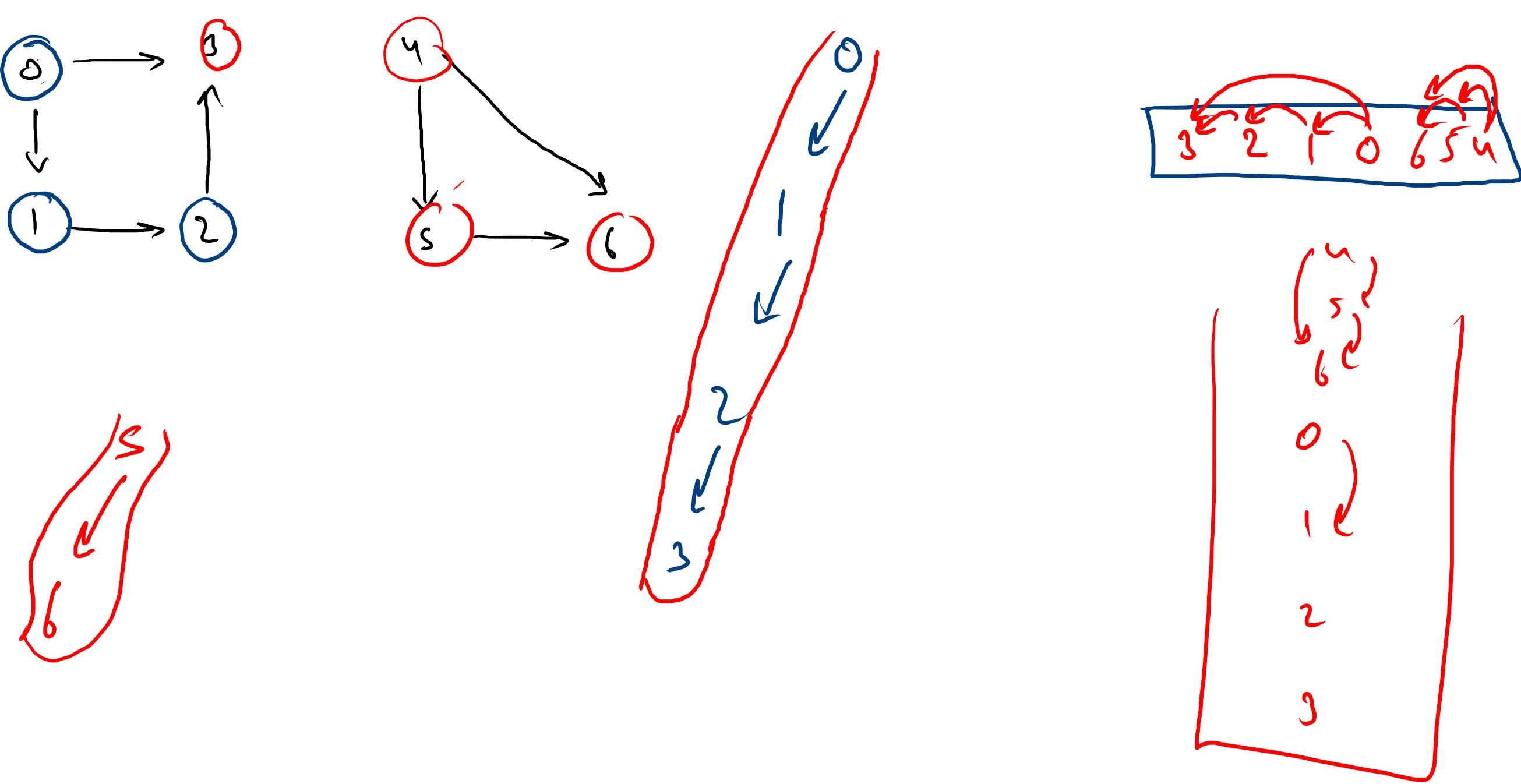
```

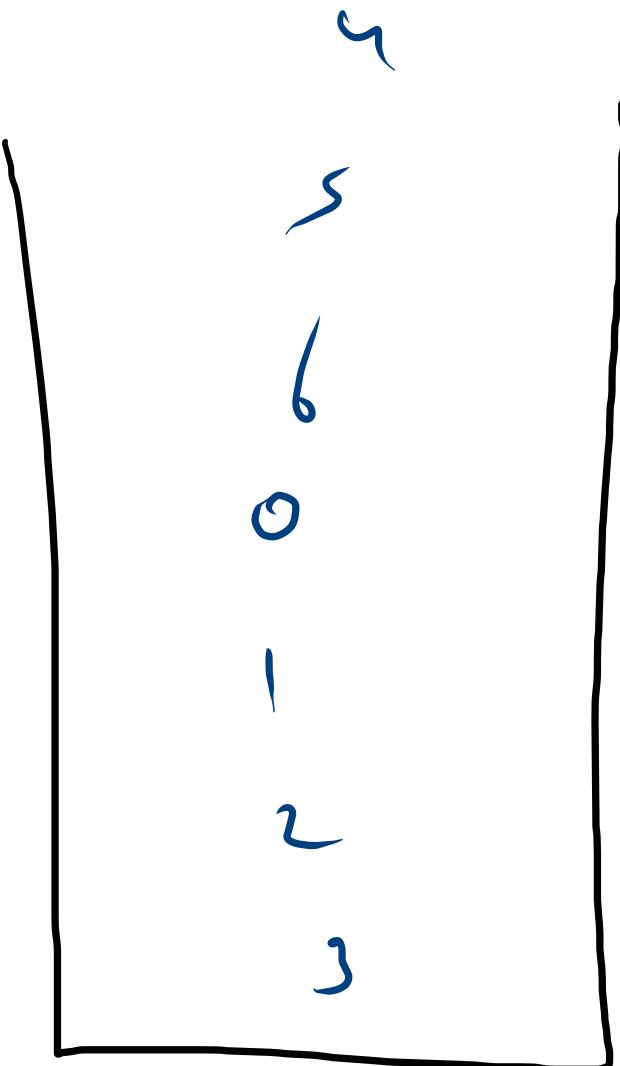
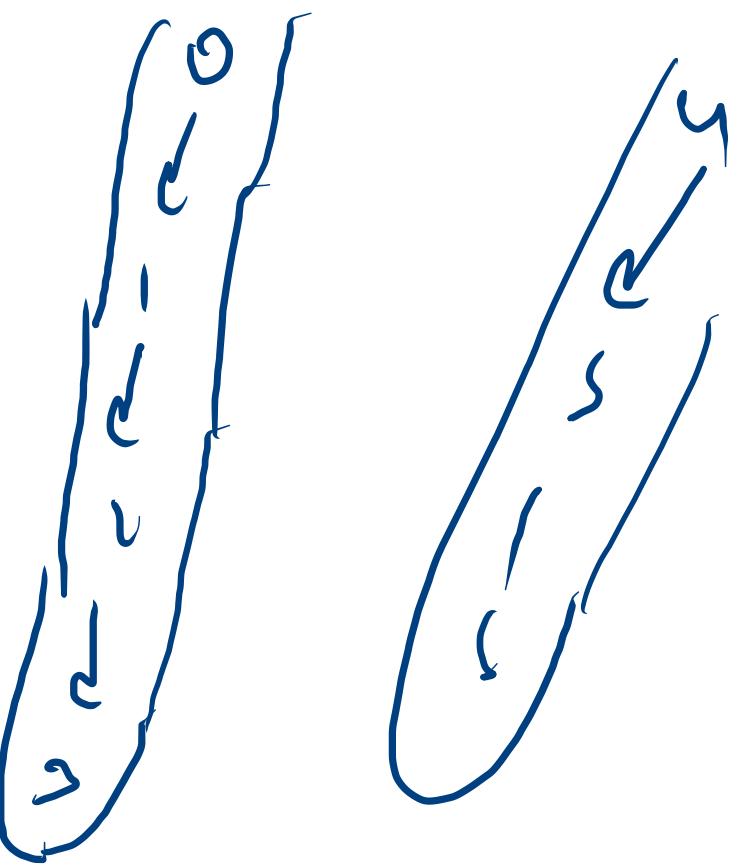
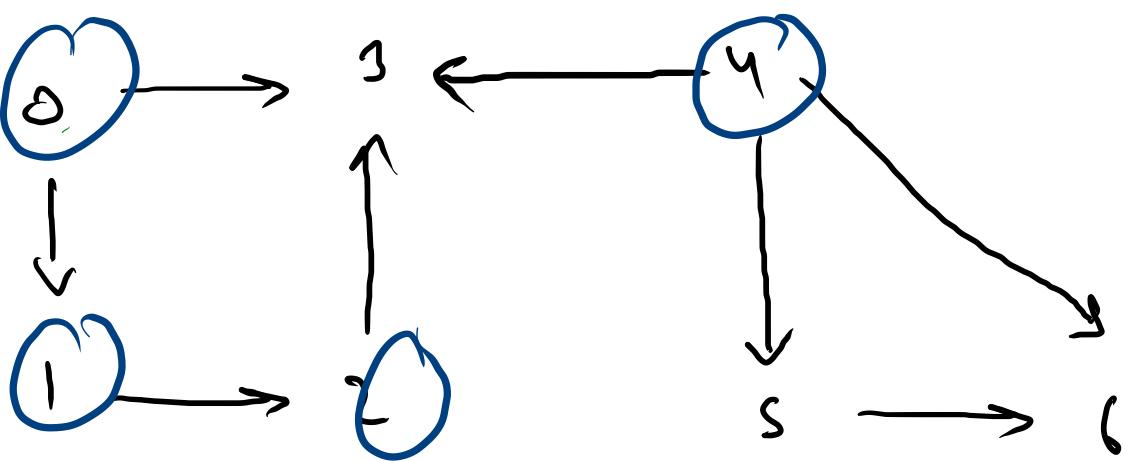
src nbr ✓  
 \_\_\_\_\_  
 0 1 0 1 0

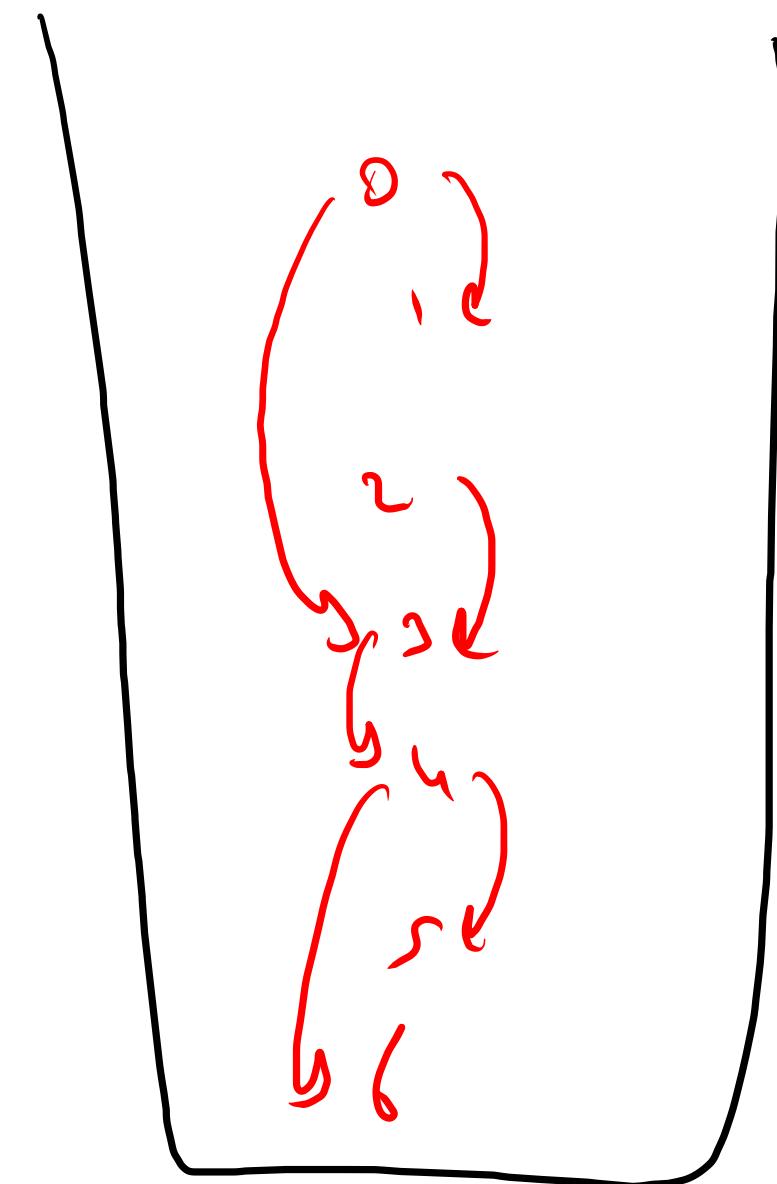
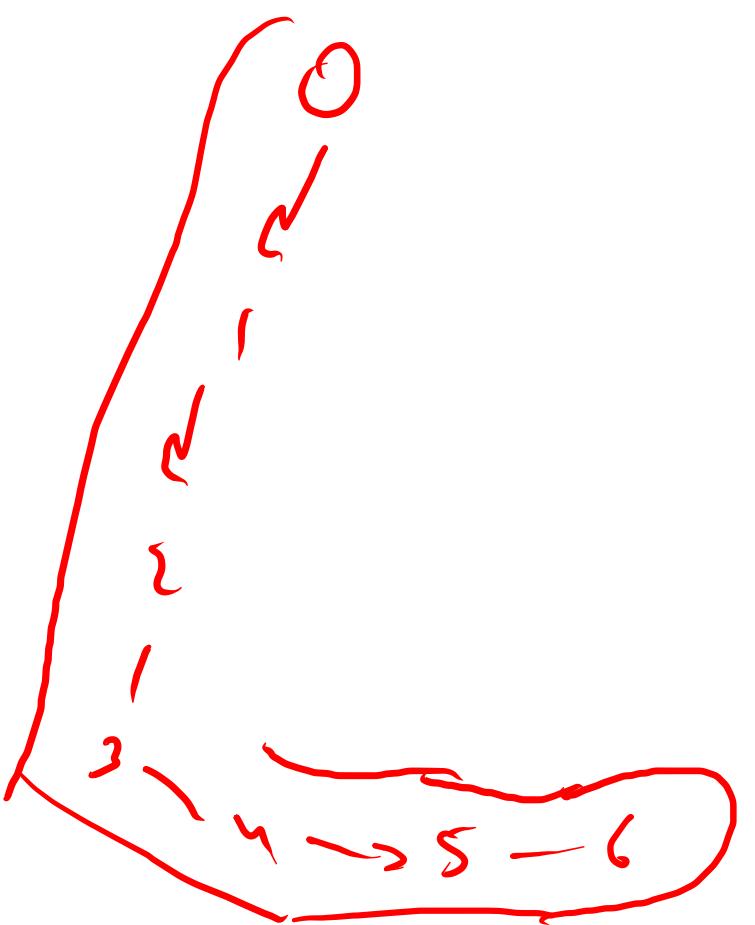
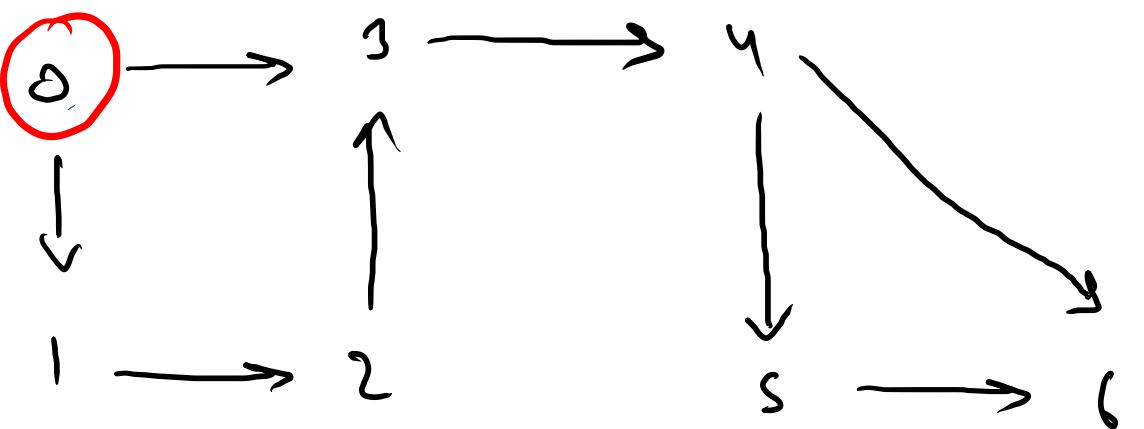
10

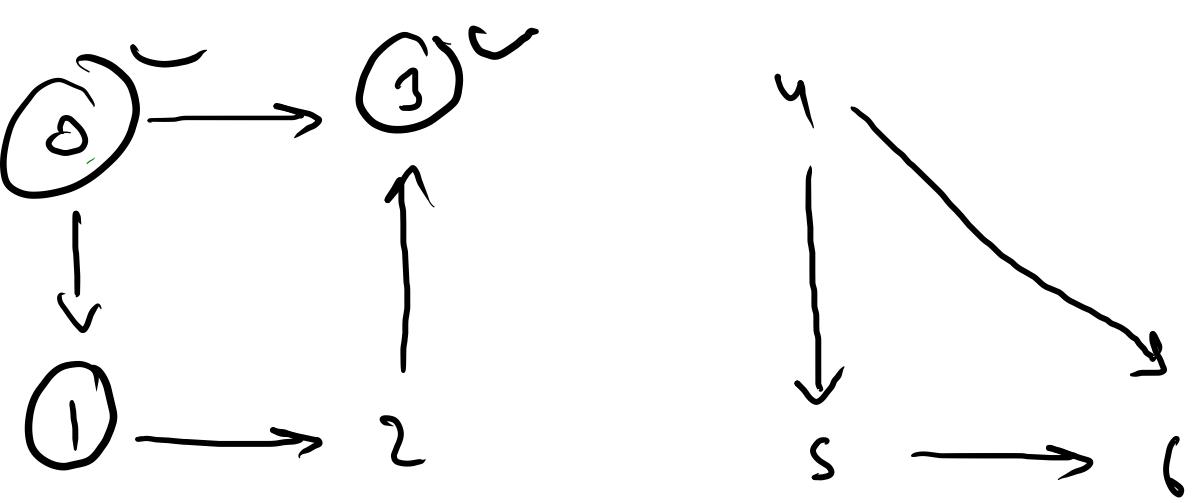


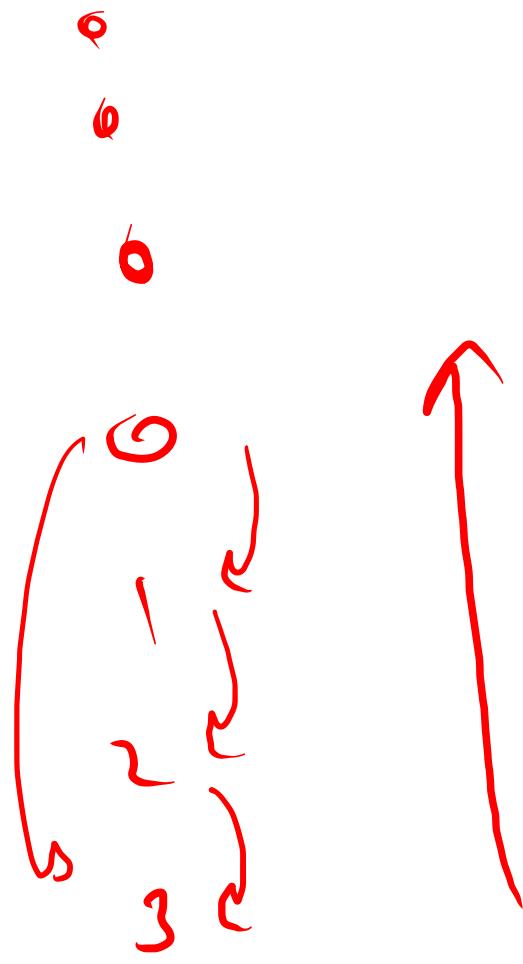
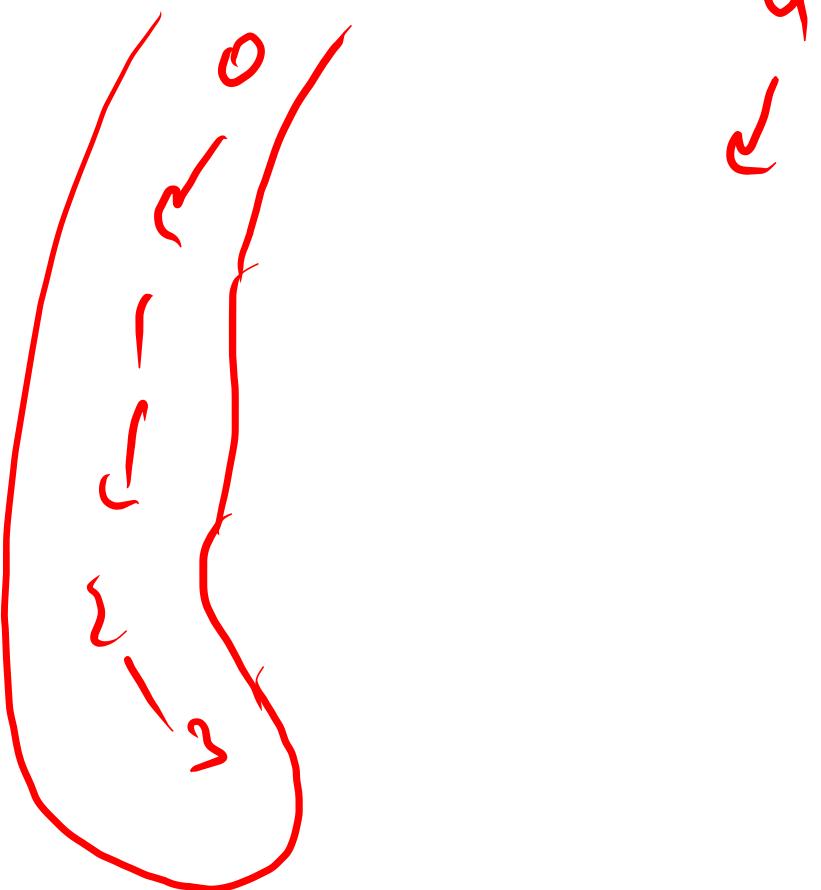
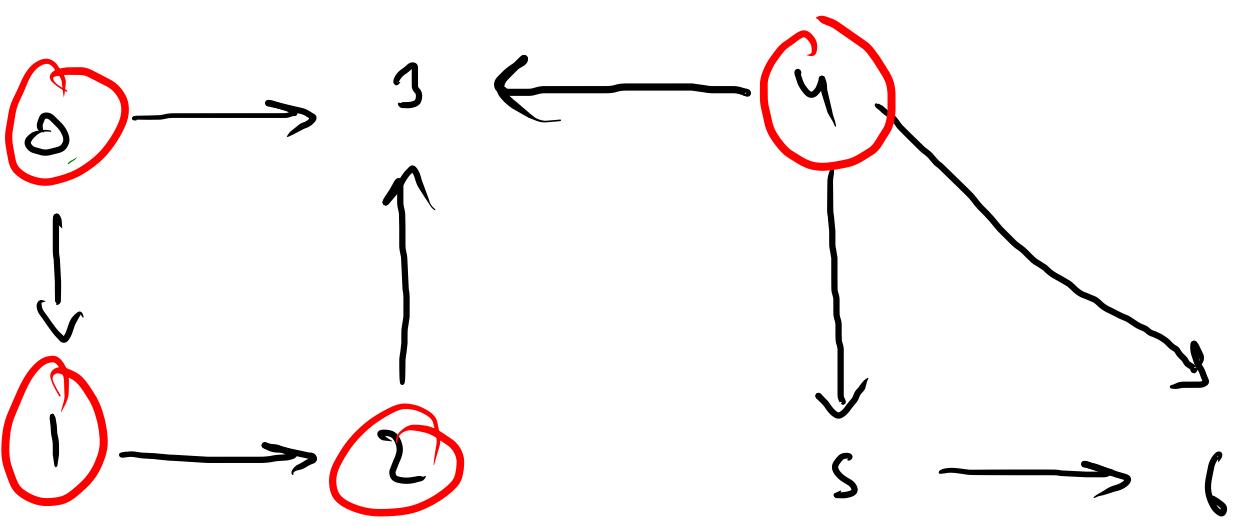
4  
5  
6  
0  
1  
2  
3

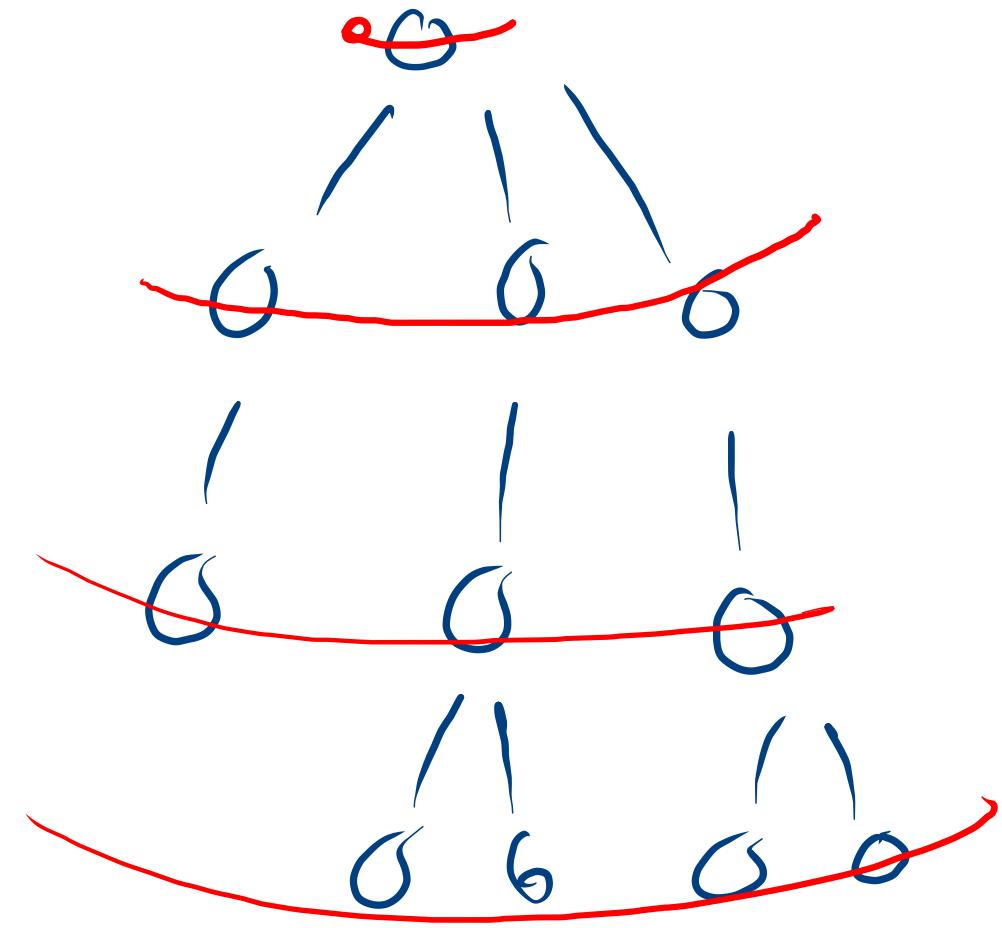
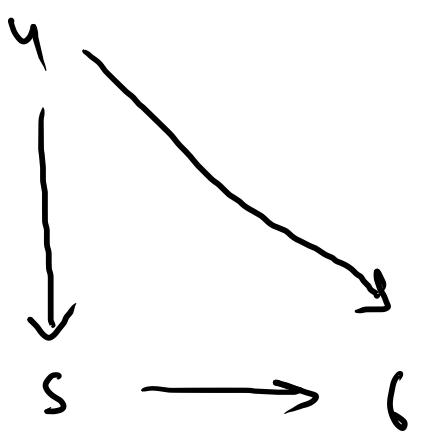
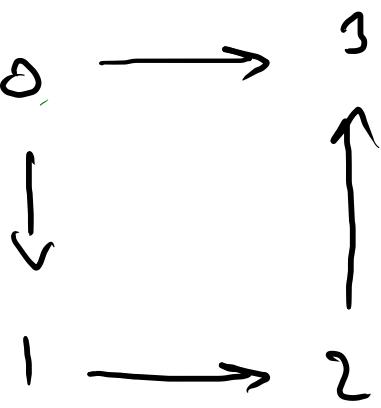






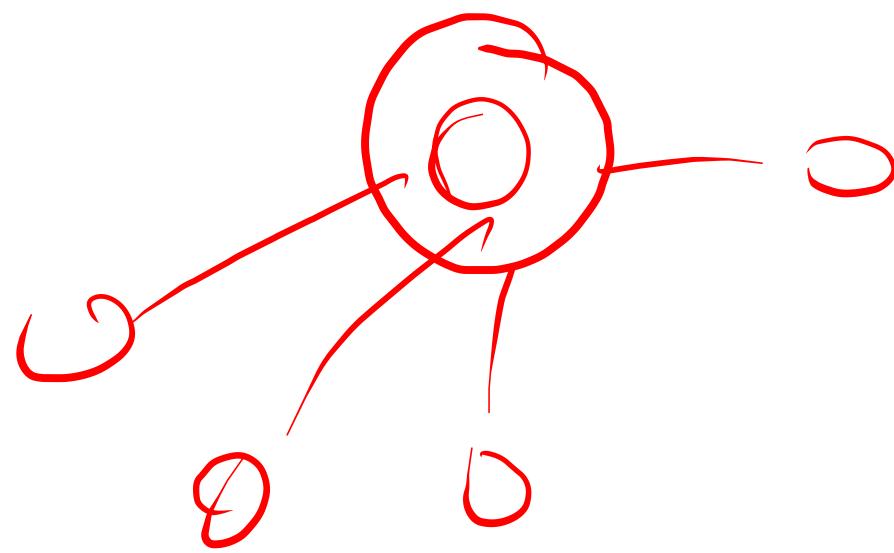
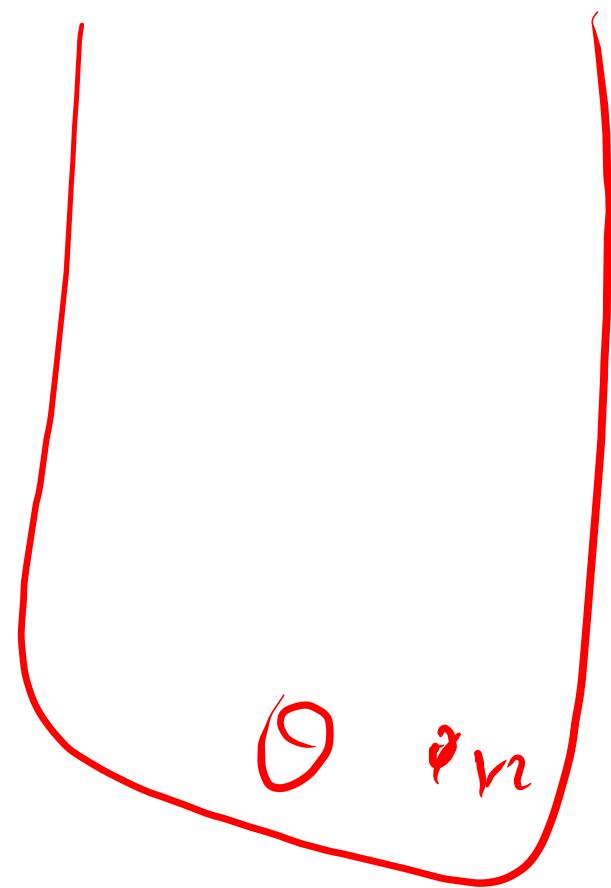


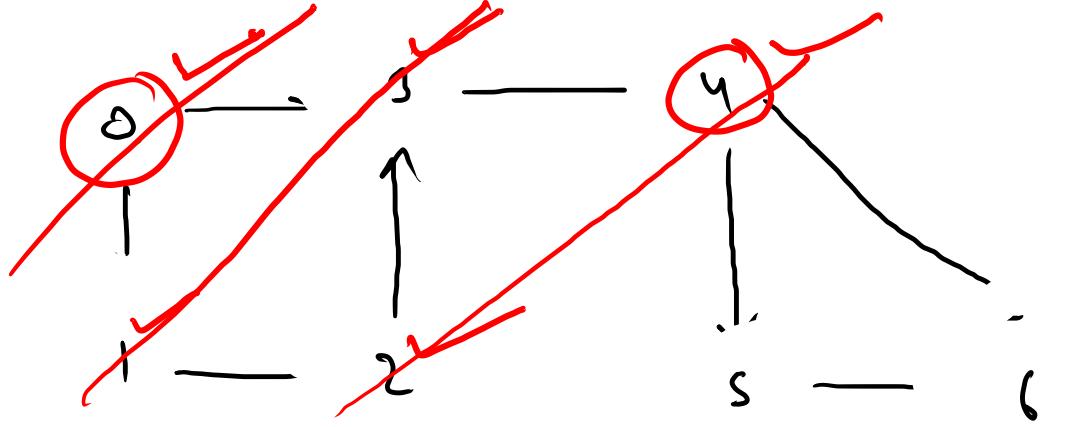




recursion

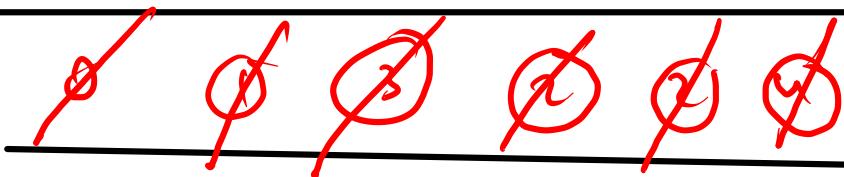
iterative stack

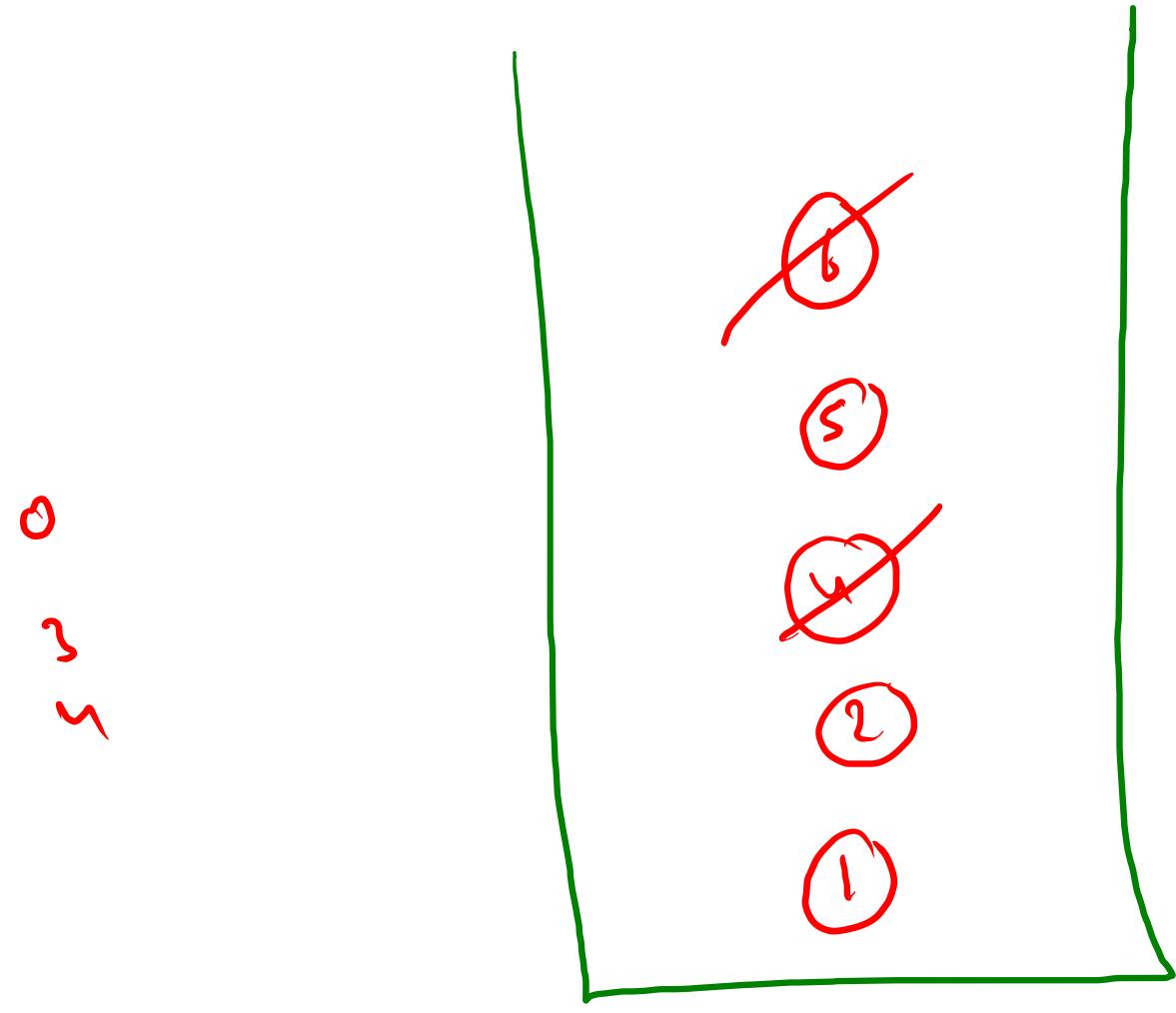
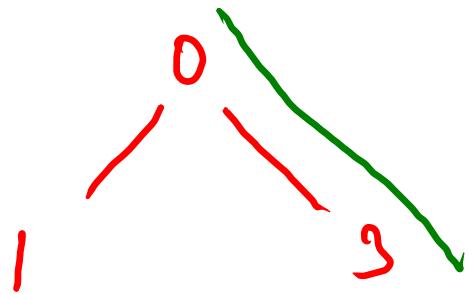
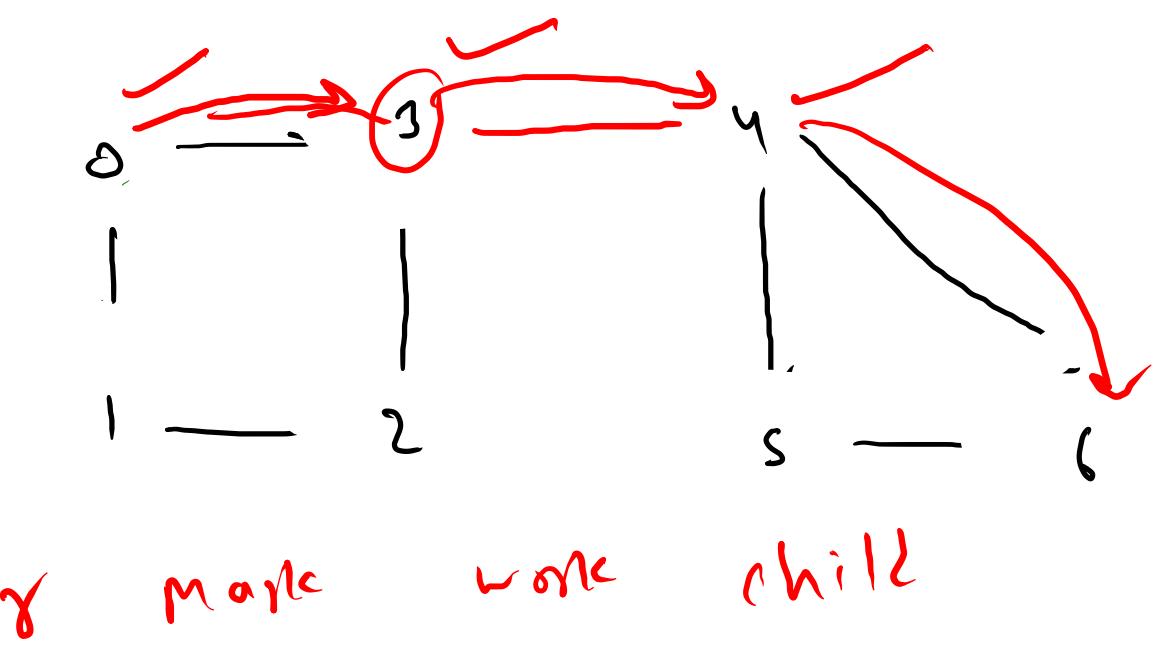


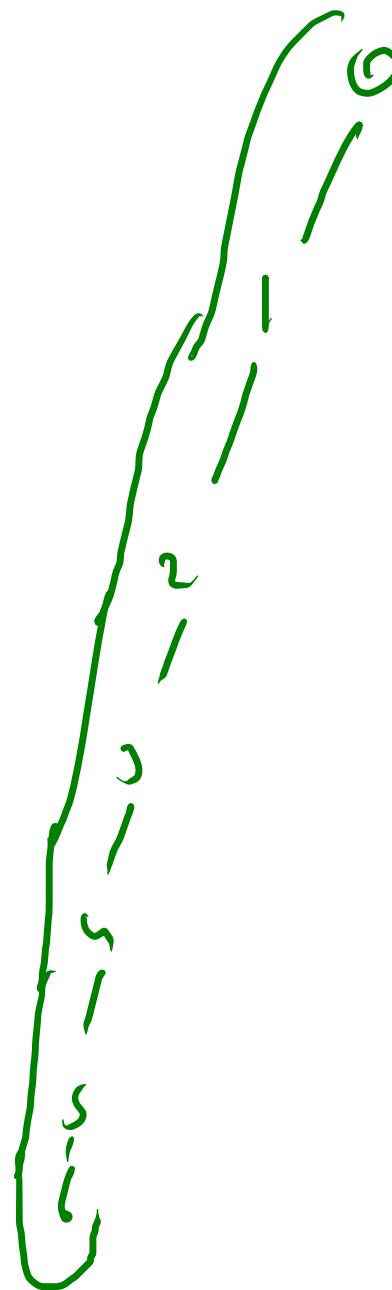
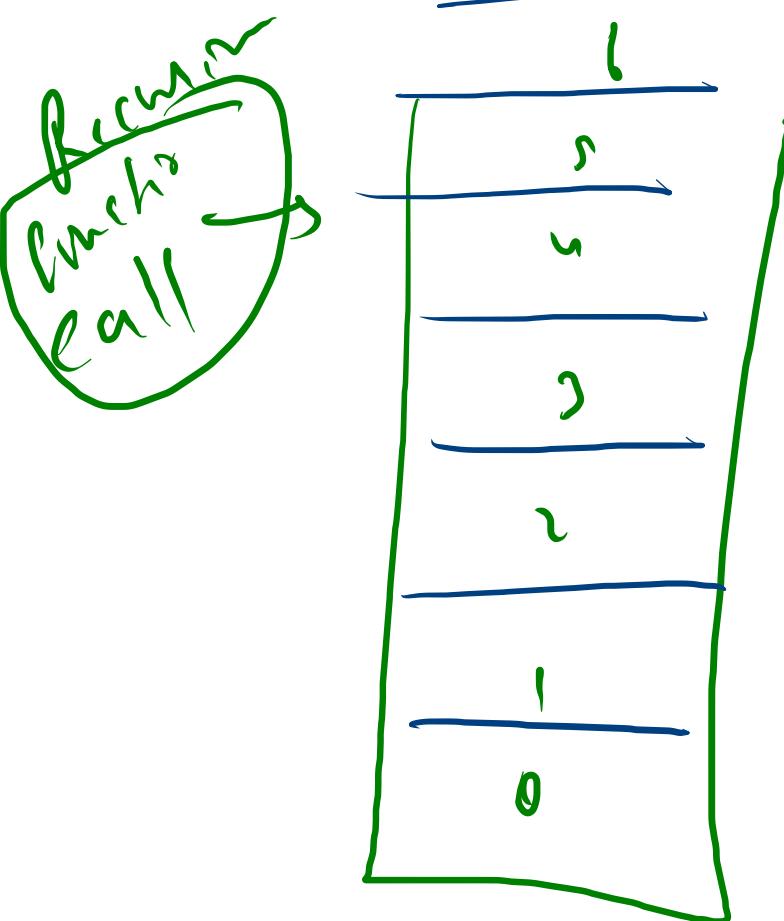
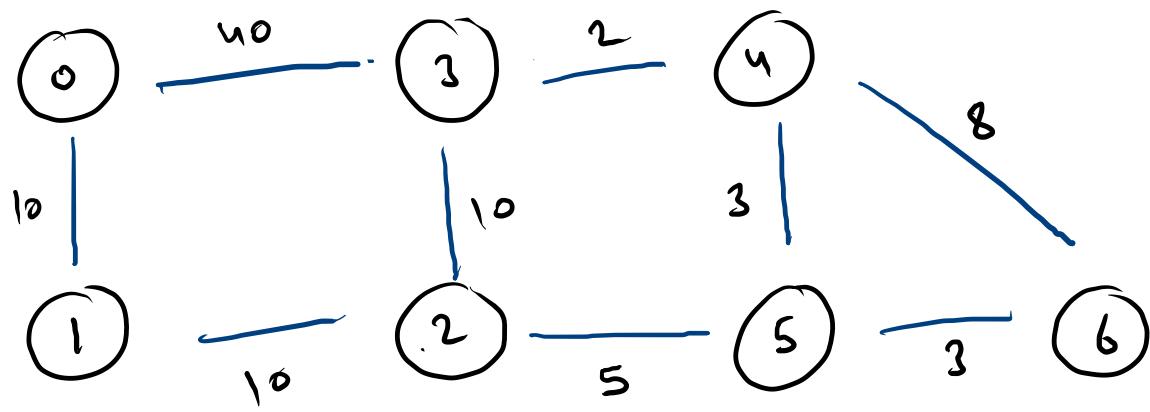


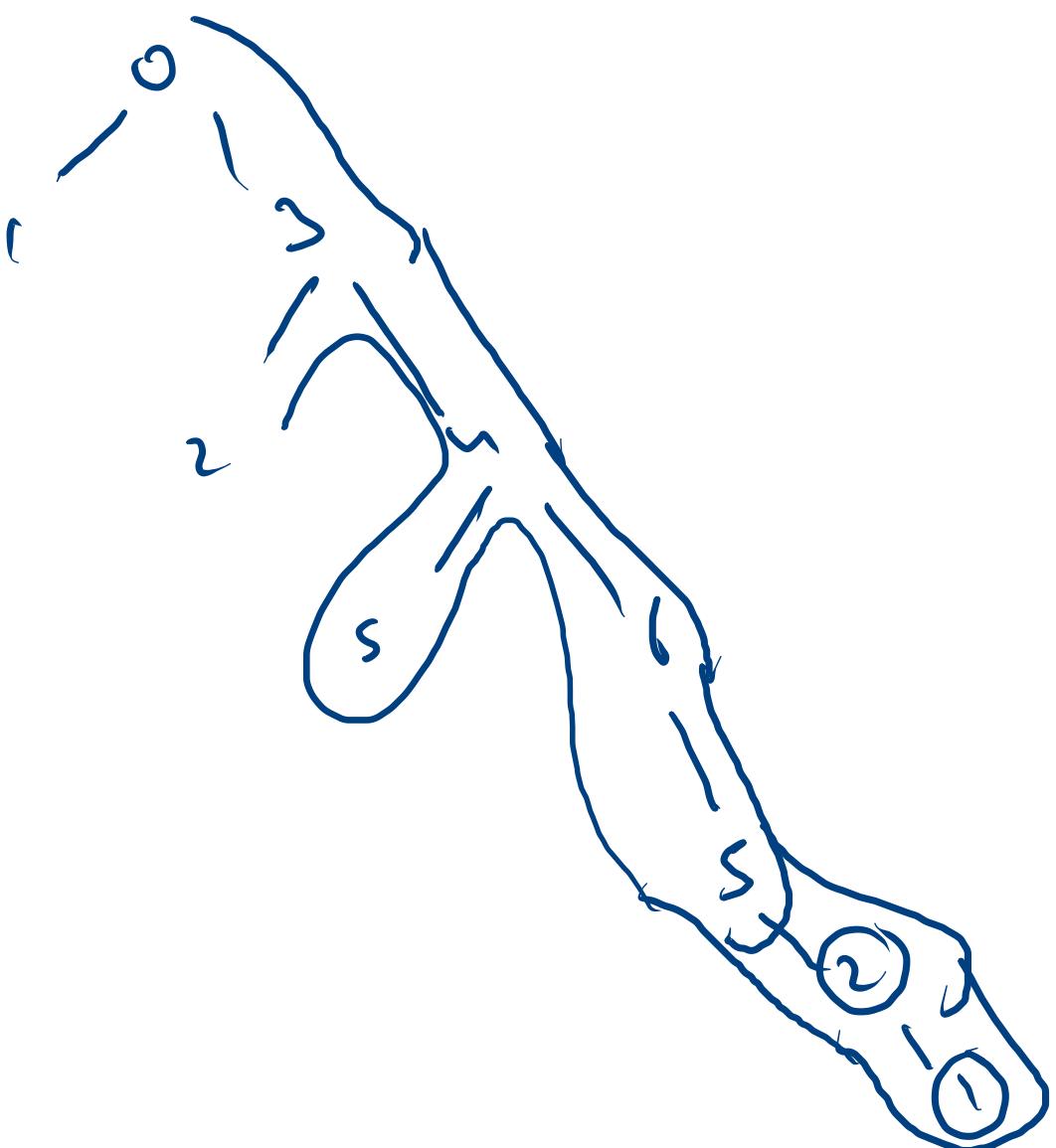
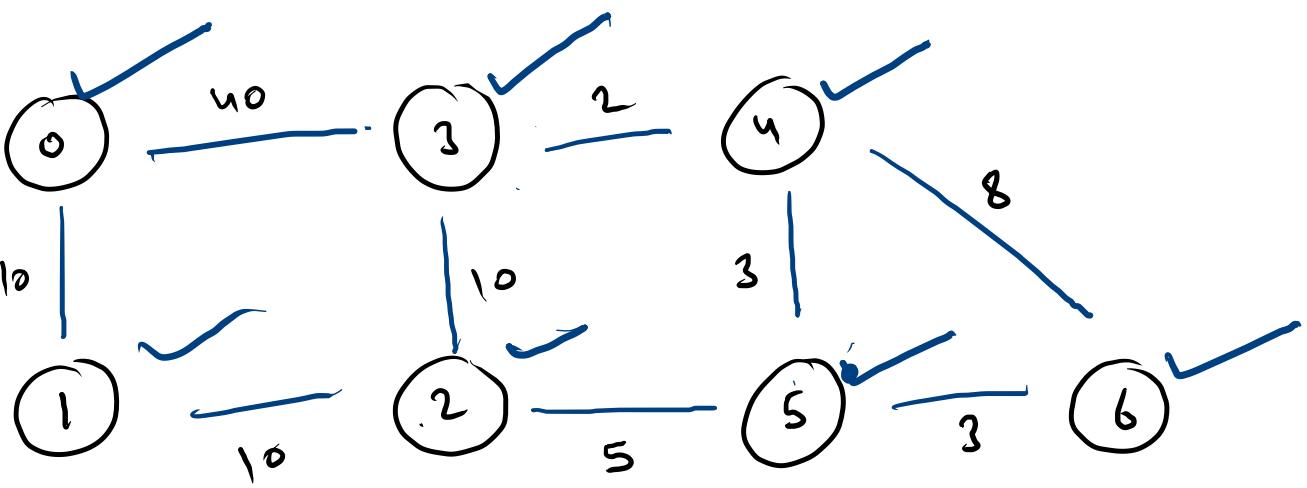
$\times$  marble wave (chi)

0  
1  
2  
3

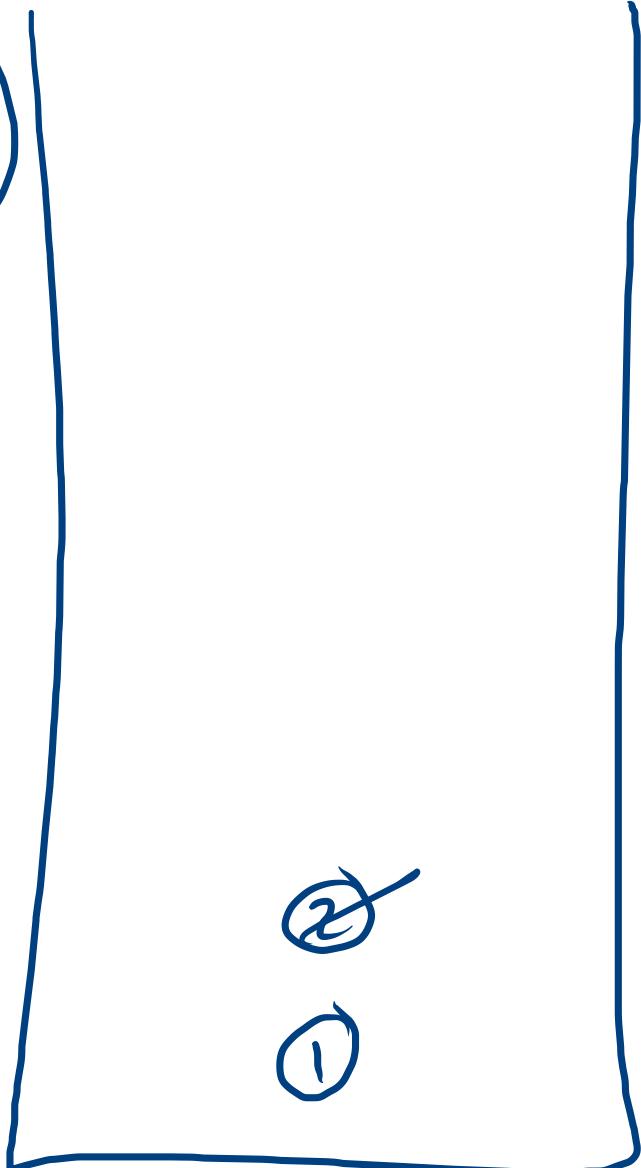


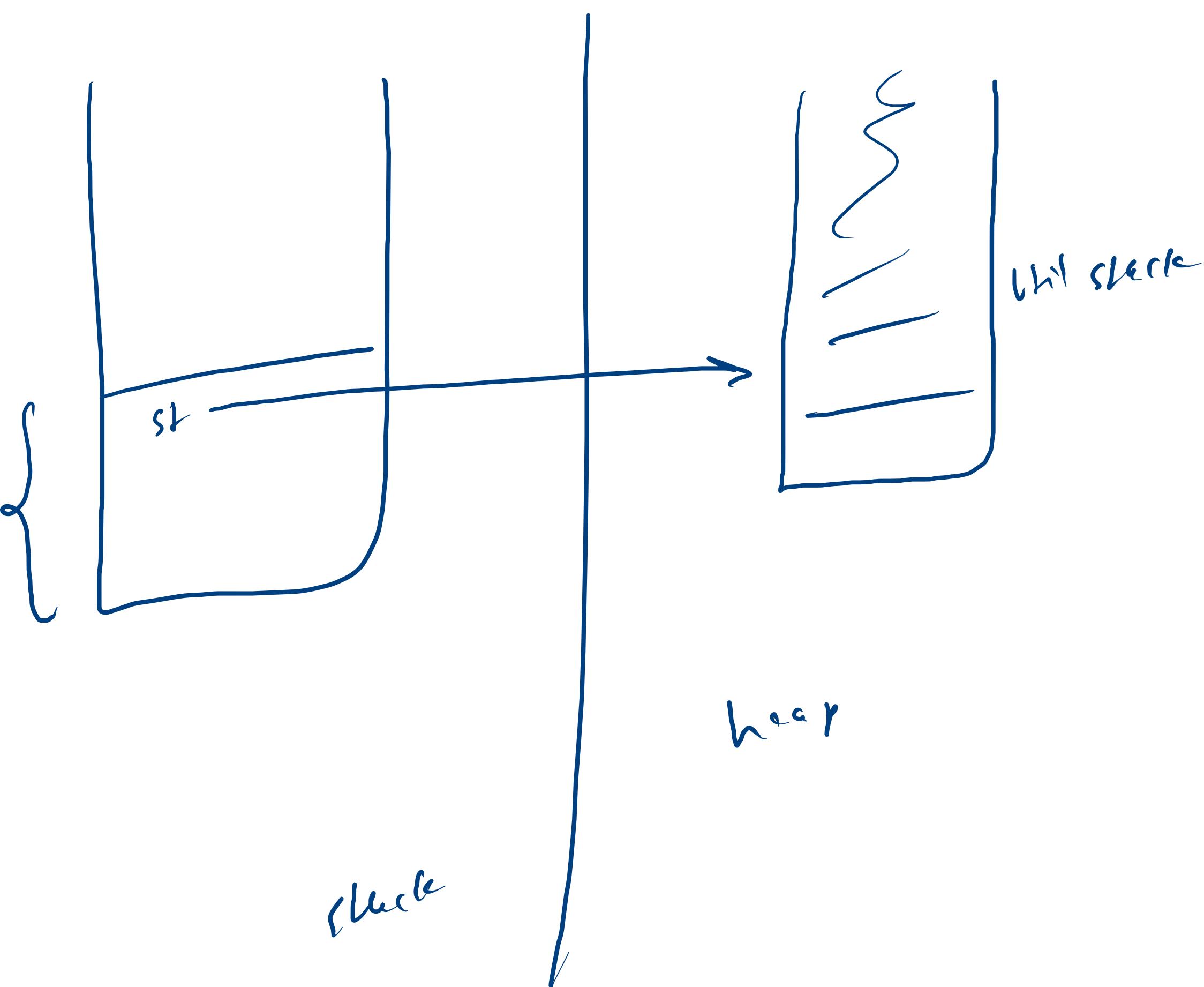


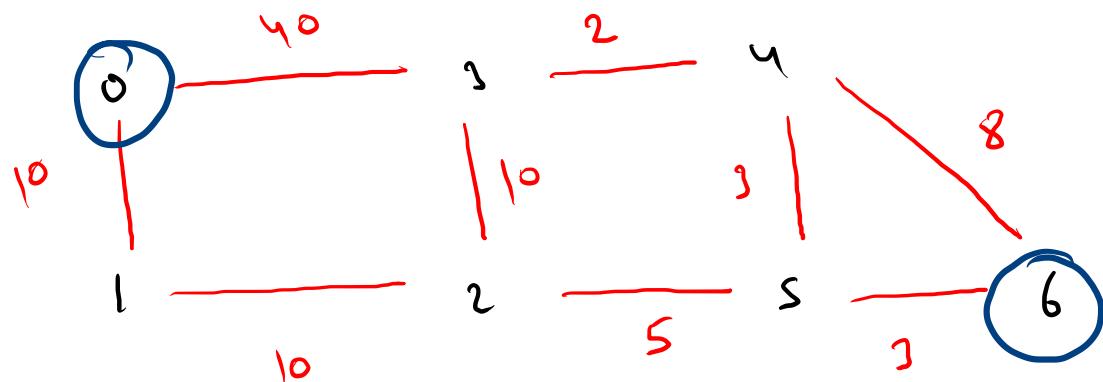




unk  
stack







- Smallest Path = 01256@28
  - Largest Path = 032546@66
  - Just Larger Path than 30 = 012546@36
  - Just Smaller Path than 30 = 01256@28
  - 4th largest path = 03456@48
- f

$s_r \leftarrow 0$

$d_v \rightarrow 6$

2 9 3 8 1 2 12 6 14 4 32 0 7 19 8 12 6

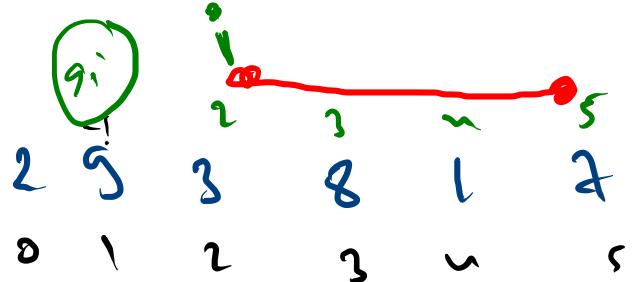
↓  
9 9 8 12 12 14 14 32

8 3  
3 2  
9 1

i

for the array [2 9 3 8 1 7 12 6 14 4 32 0 7 19 8 12 6] and k = 4, the answer is [9 9 8 12 12 14 14 32 32 32 32 19 19 19]

$k = n$



6 12 6 14 4 32 0 7 19 8 12 6  
0 1 2 3 4 5 6 7 8 9 10

hgeR

1 6 3 ~~7~~ 5 6 8

$g_i$

9 9 8 12

$i \leftarrow (g_i < i)$   
while ( $h_{g_i}[g_i] < i + k$ )  
 $g_i = h_{g_i}[g_i]$

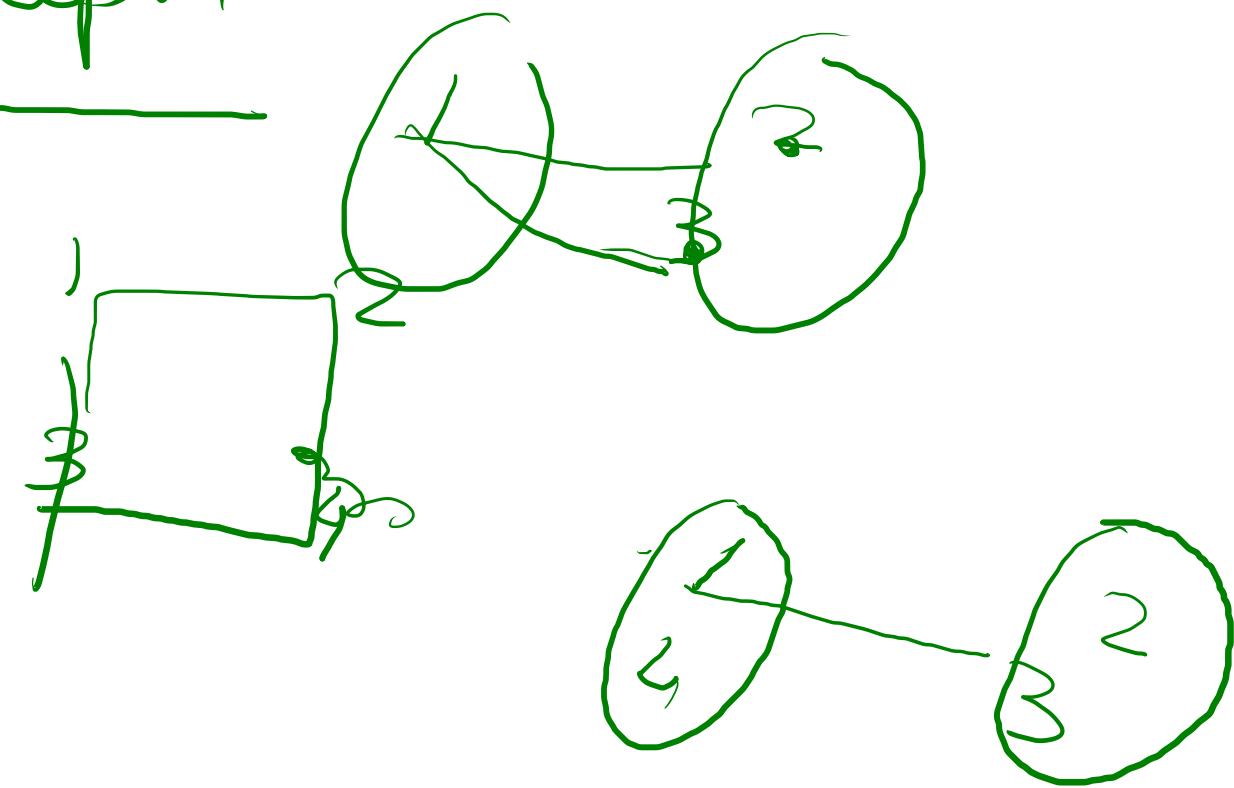
$6$

$2+4$

R  
Bipartite



Graph



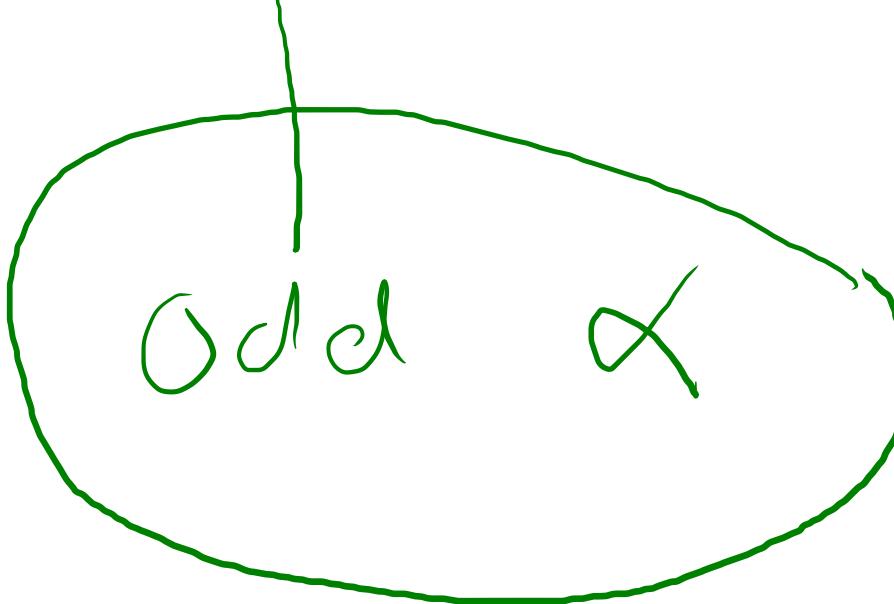
R

①

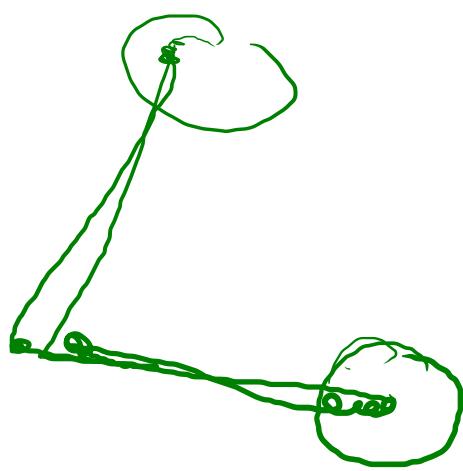
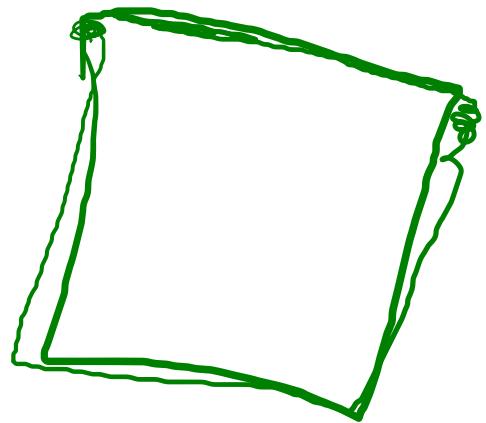
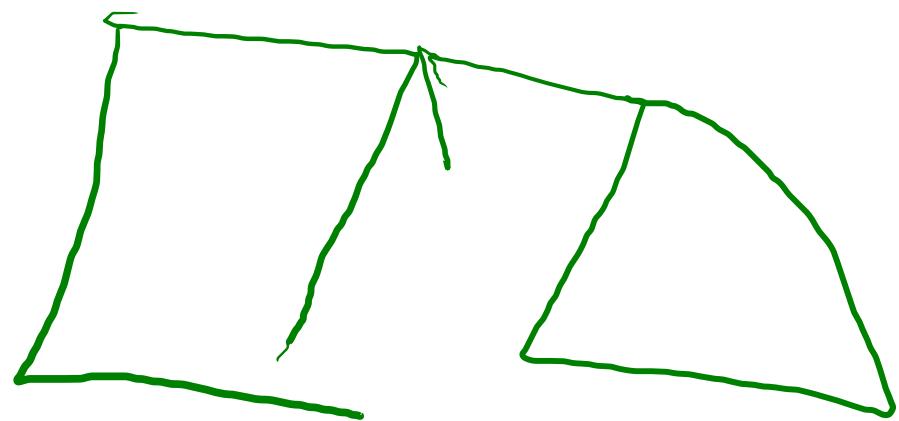
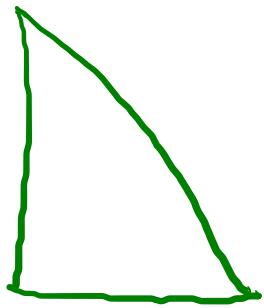
Ay Mic

②

Cy Mic Even L



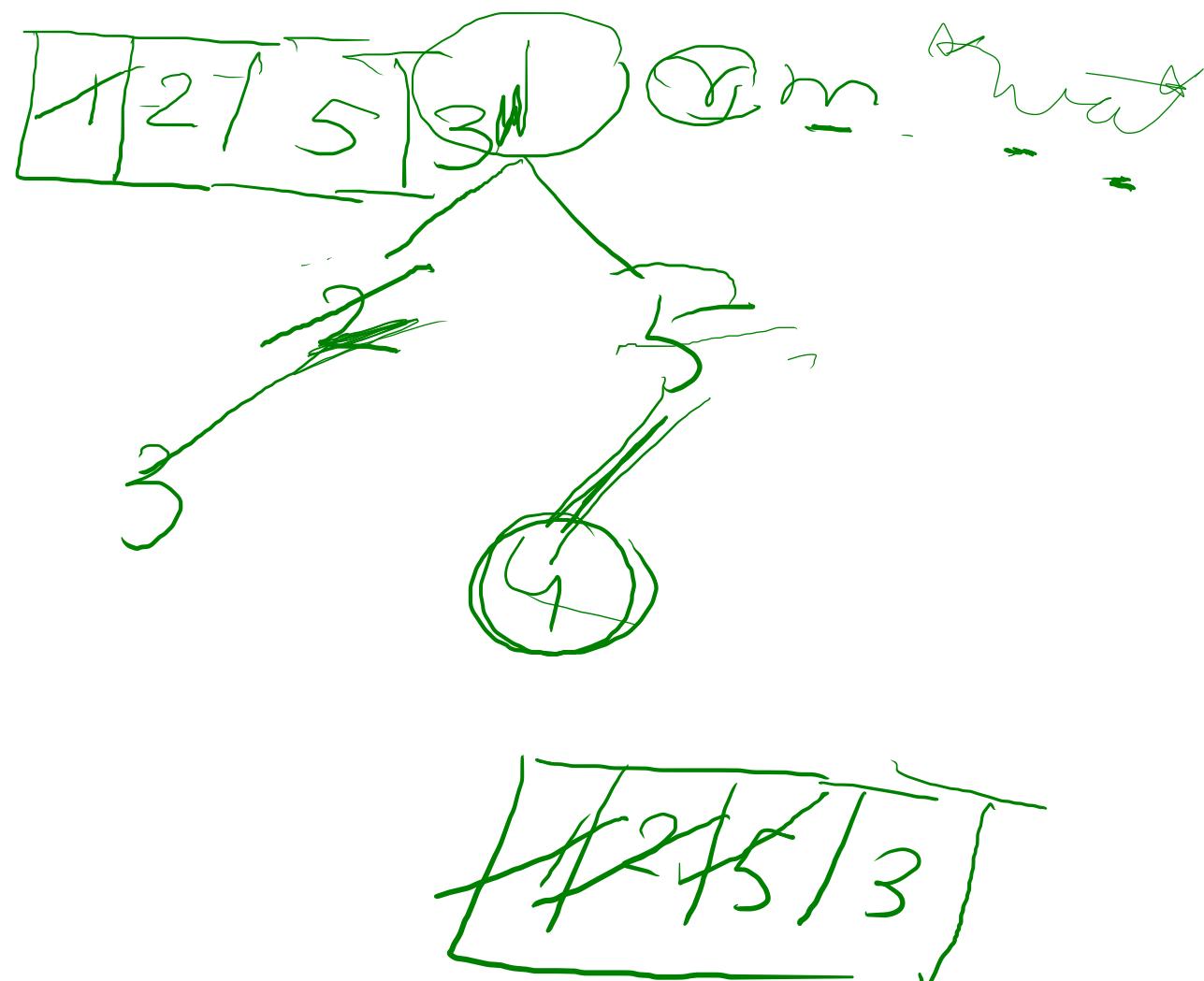
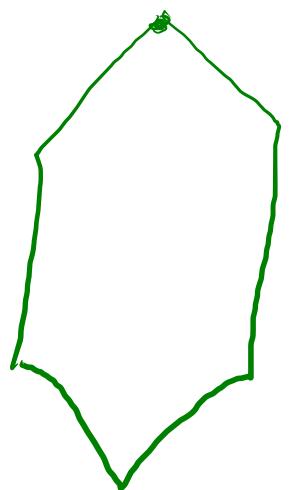
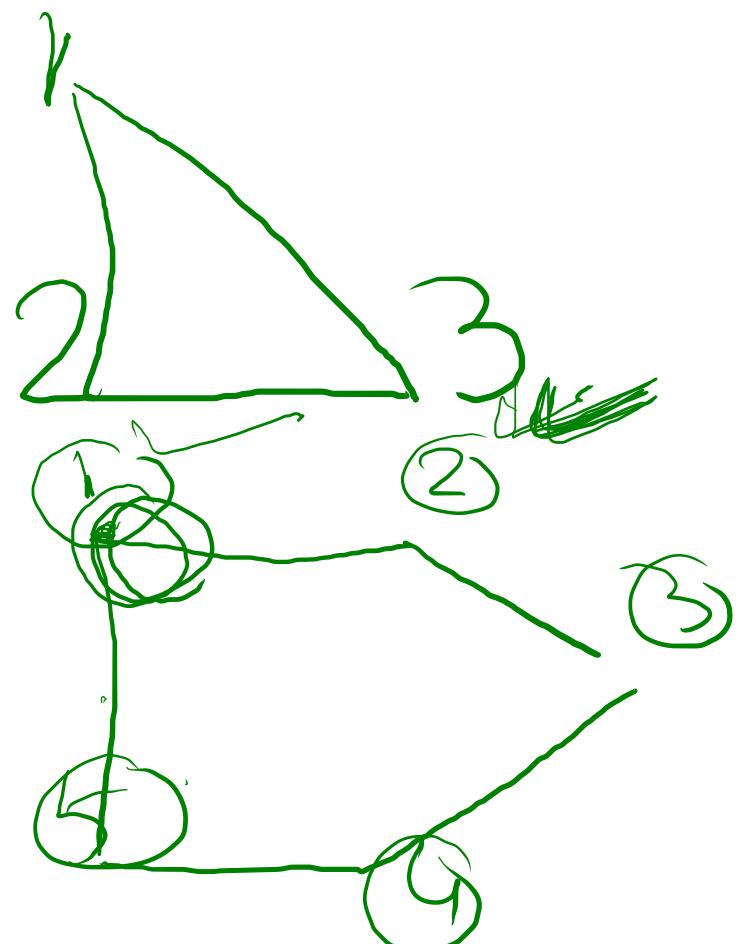
R



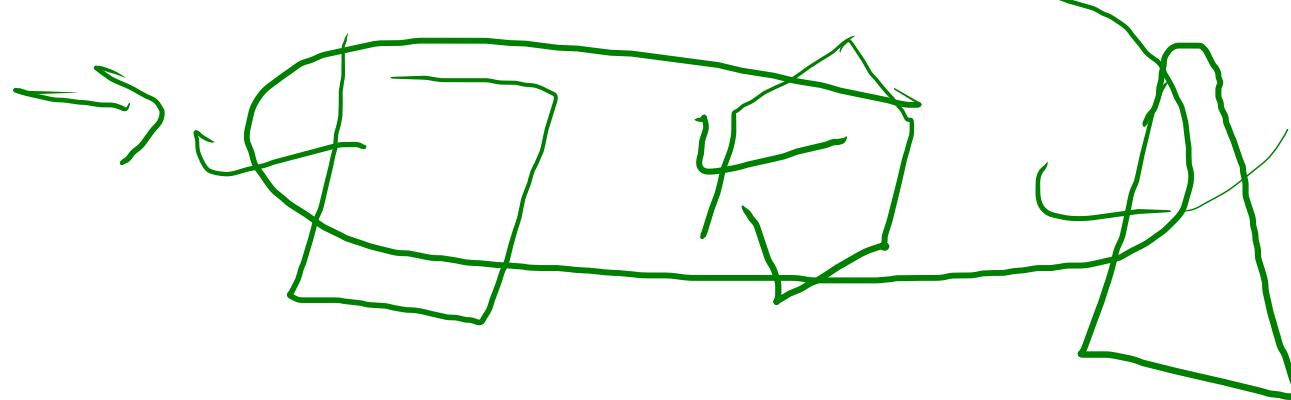
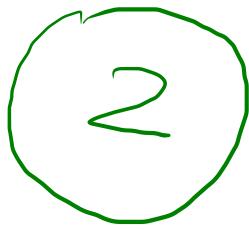
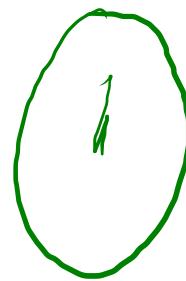
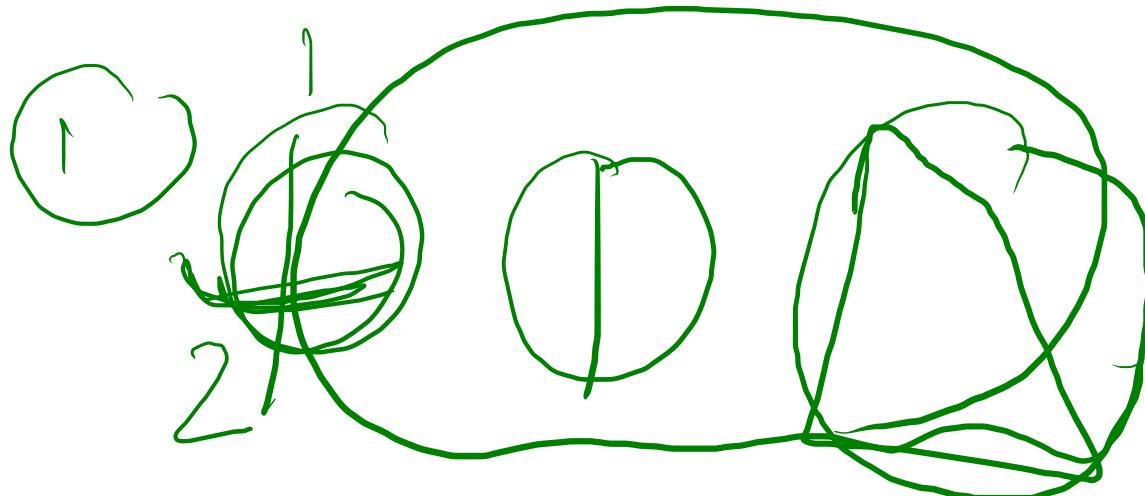
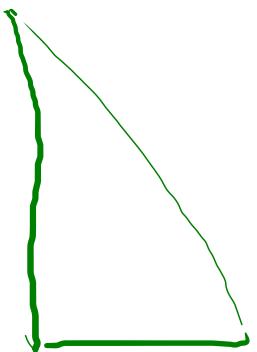


visited ✓ 3

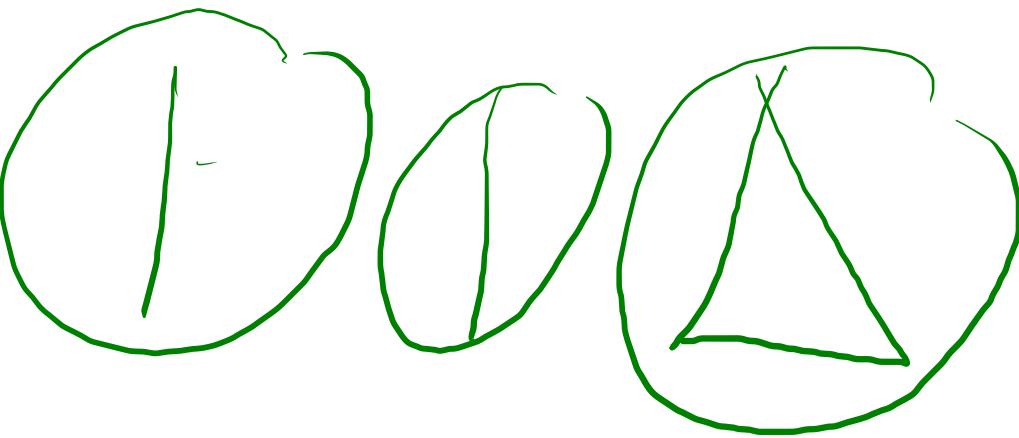
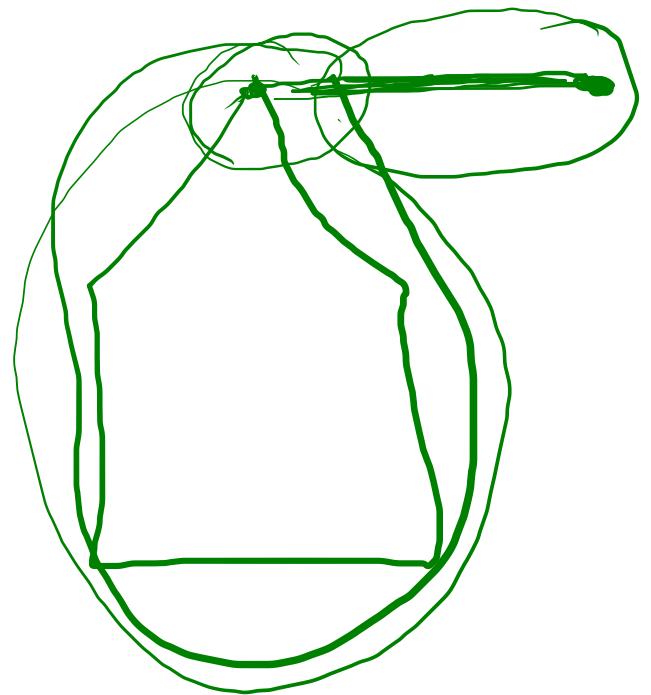
R

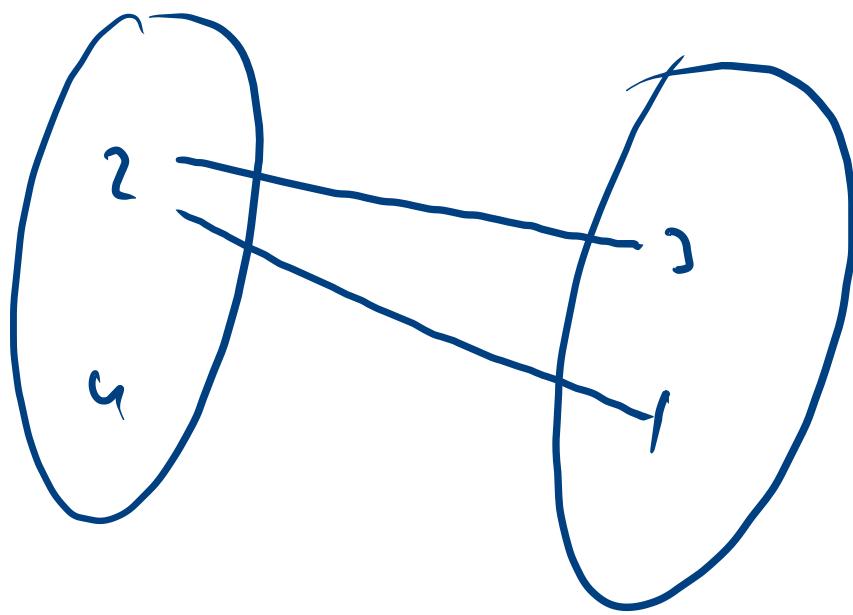
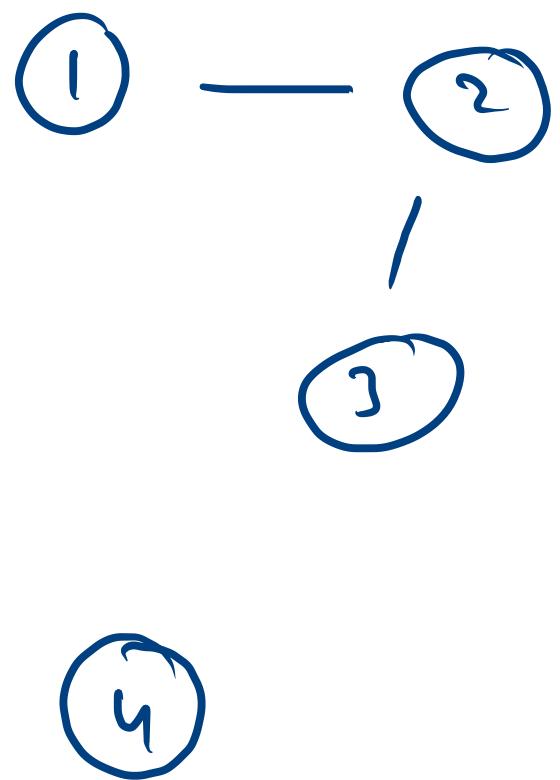


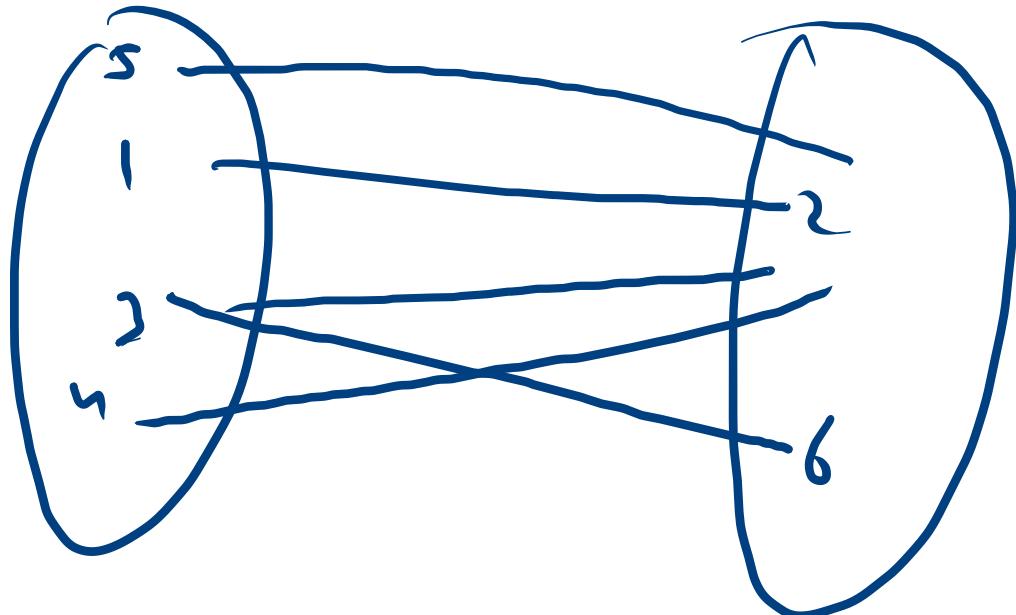
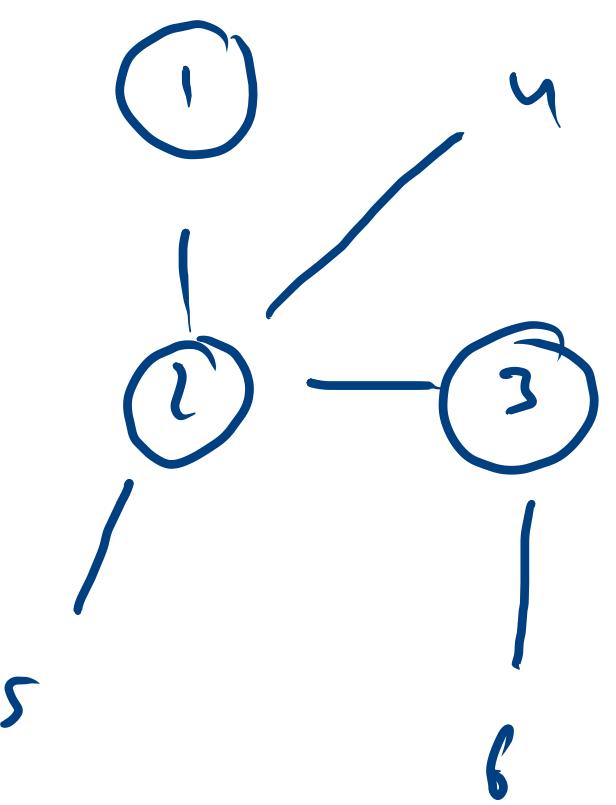
R

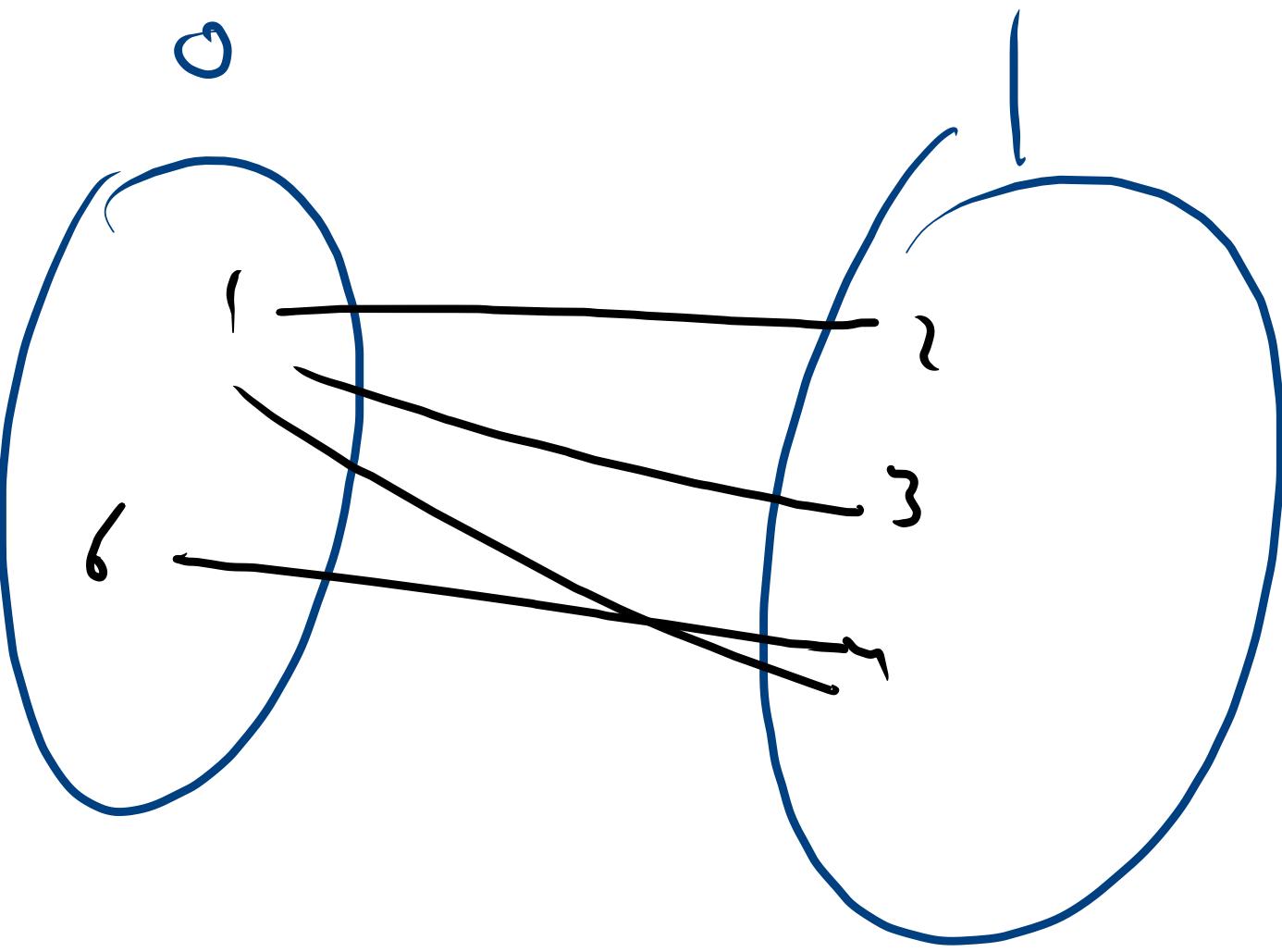
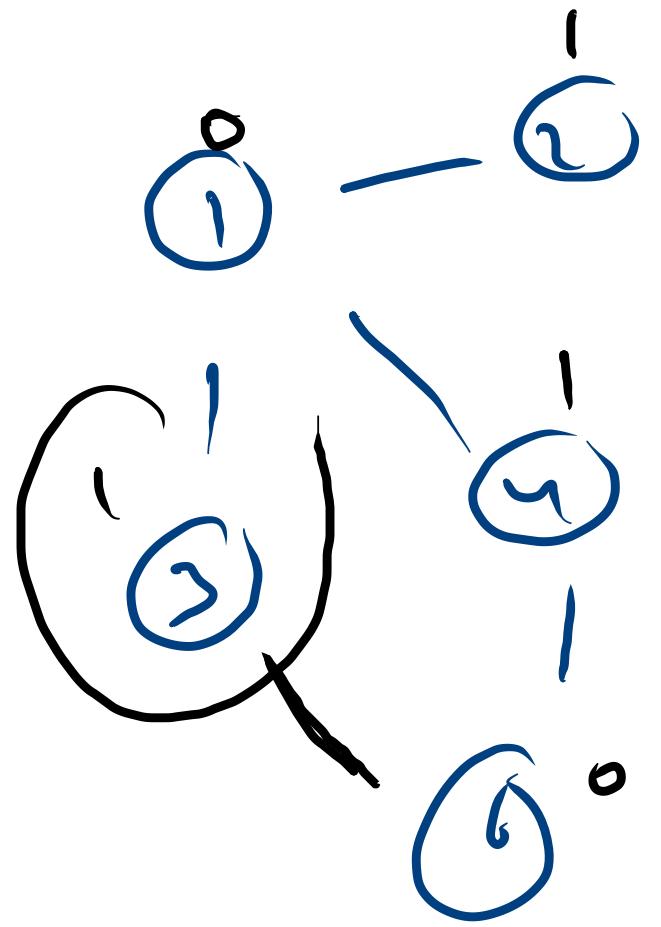


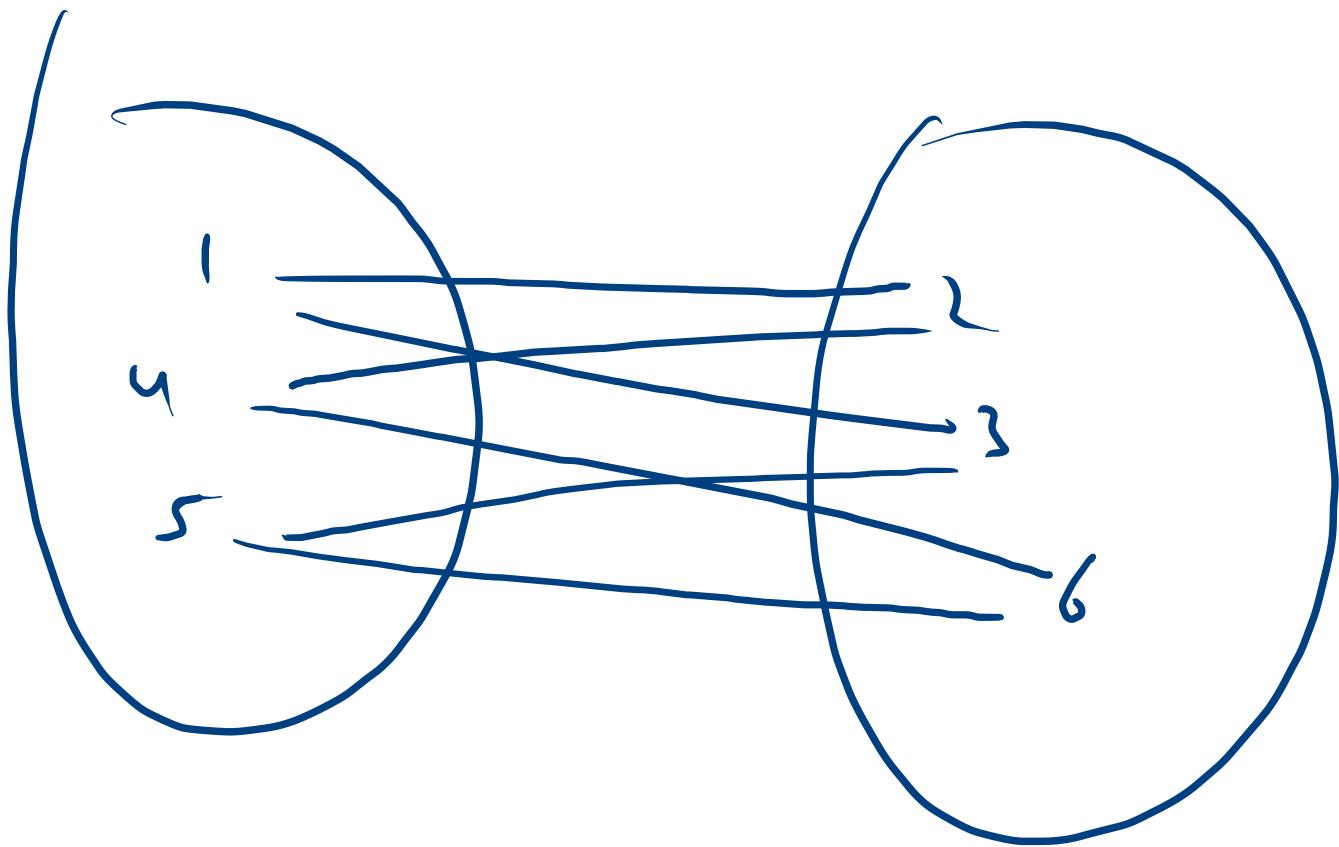
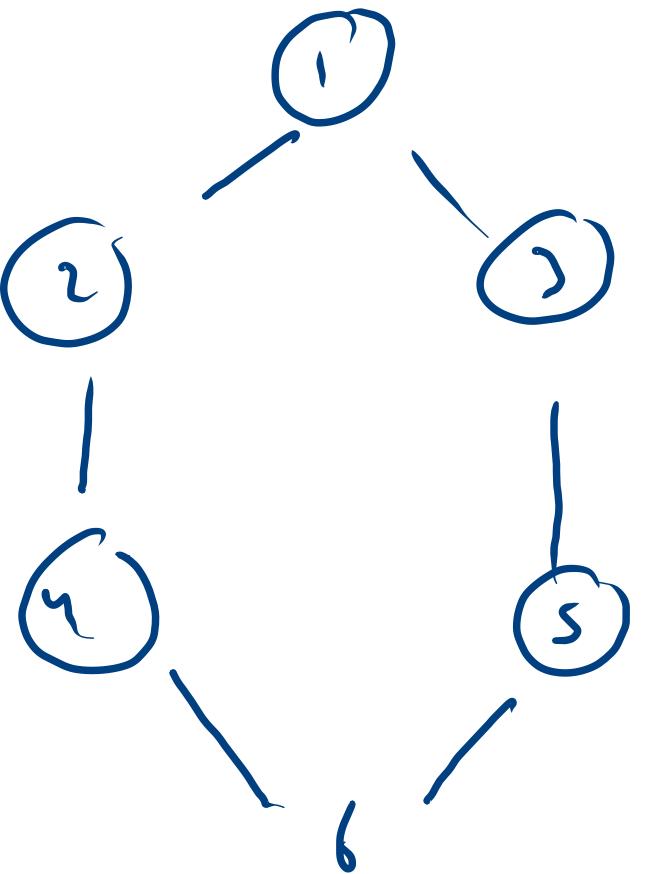
R

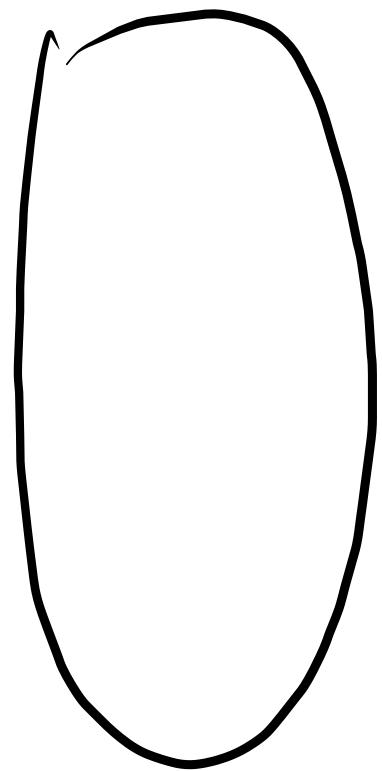
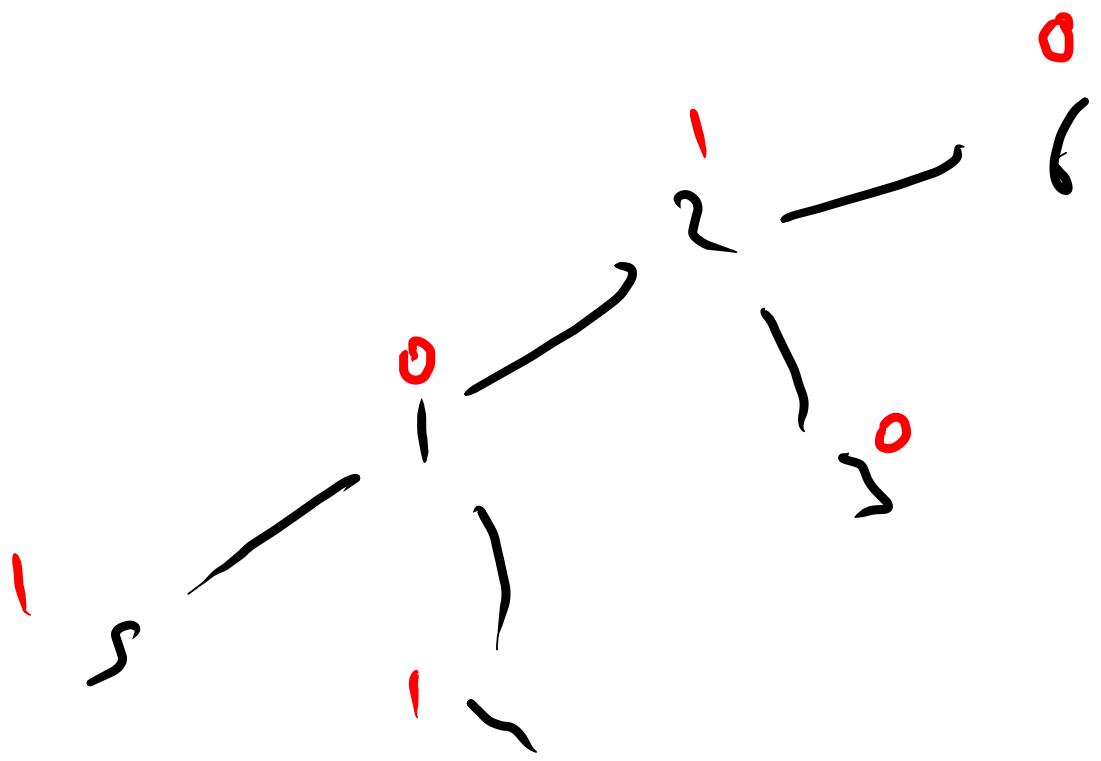
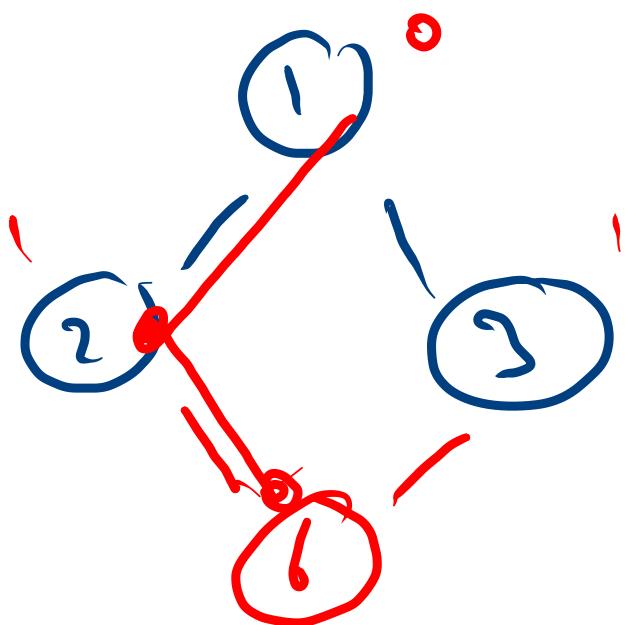


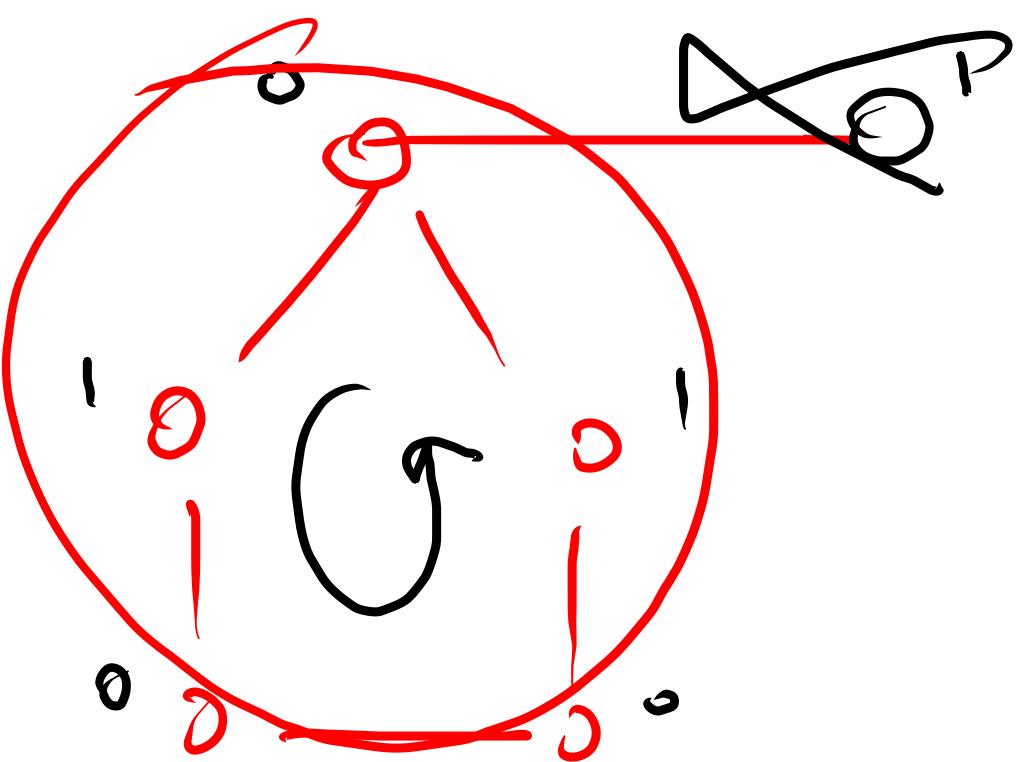


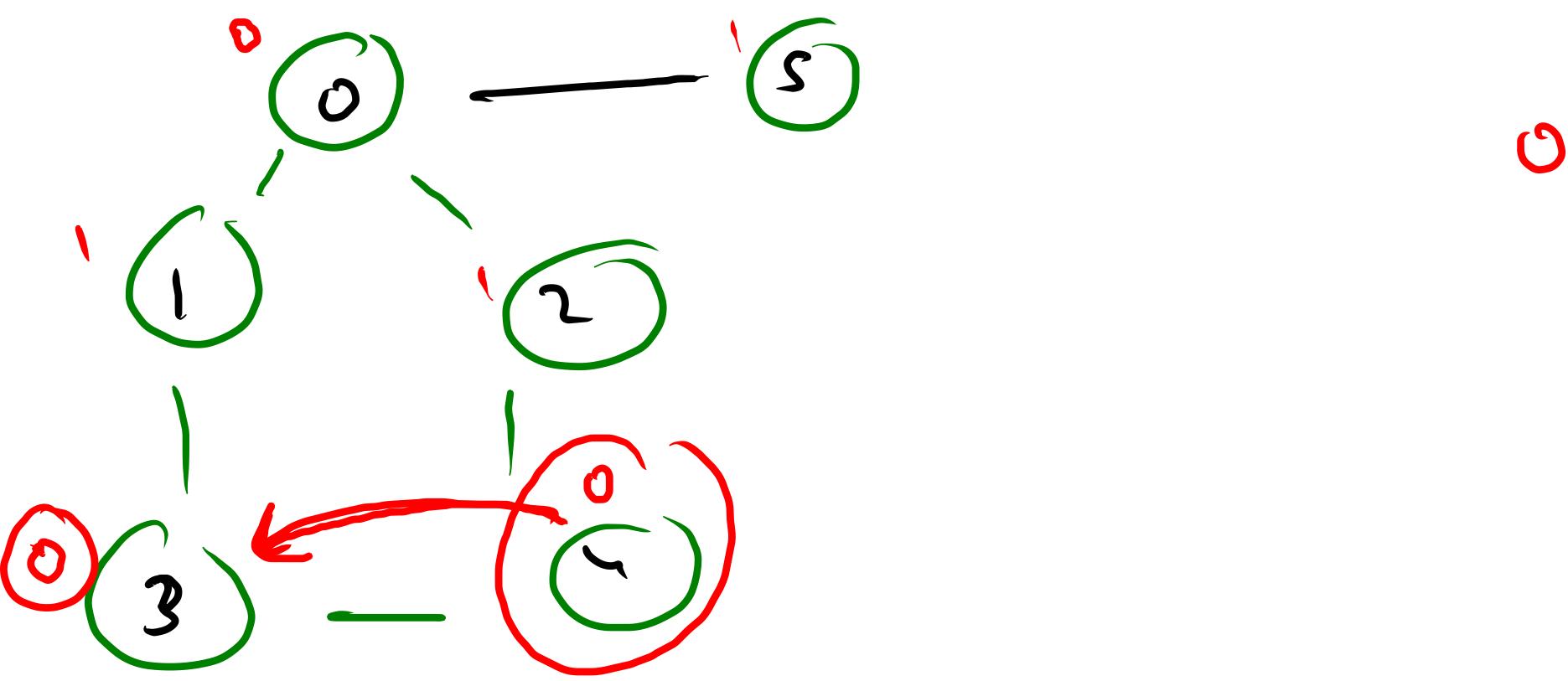


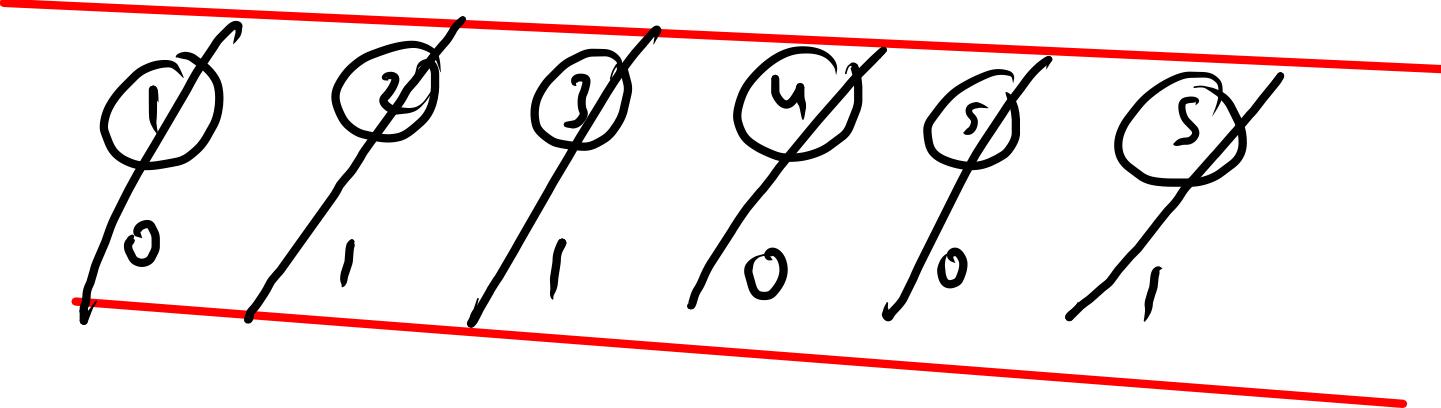
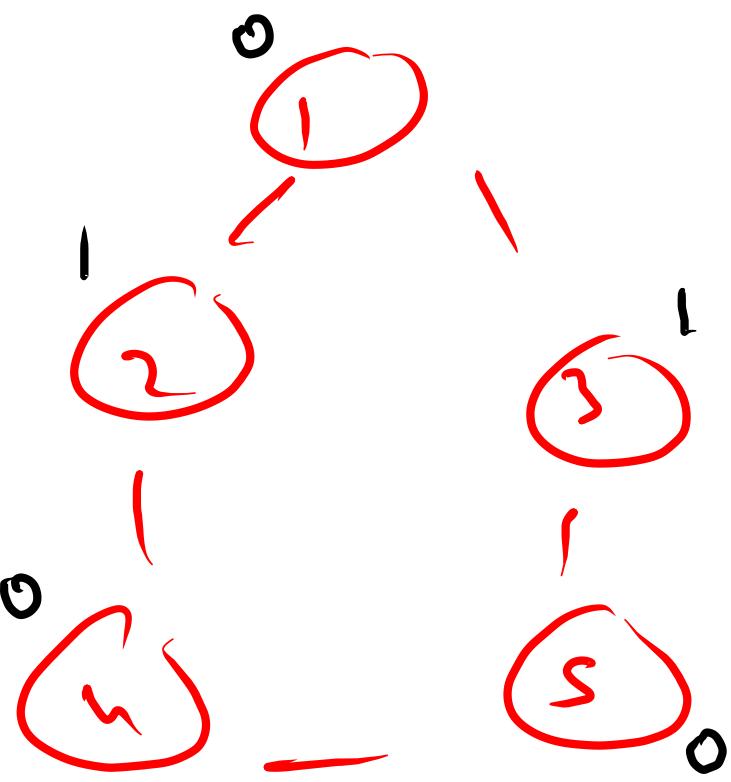


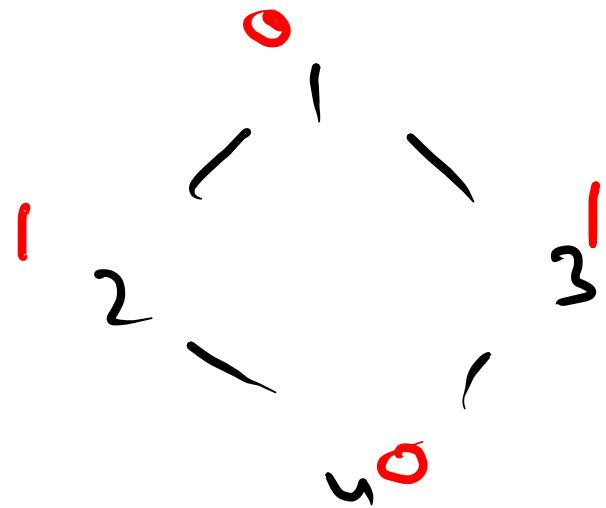












inL ar

