Self-stabilization

Algo 21

Self stabilization

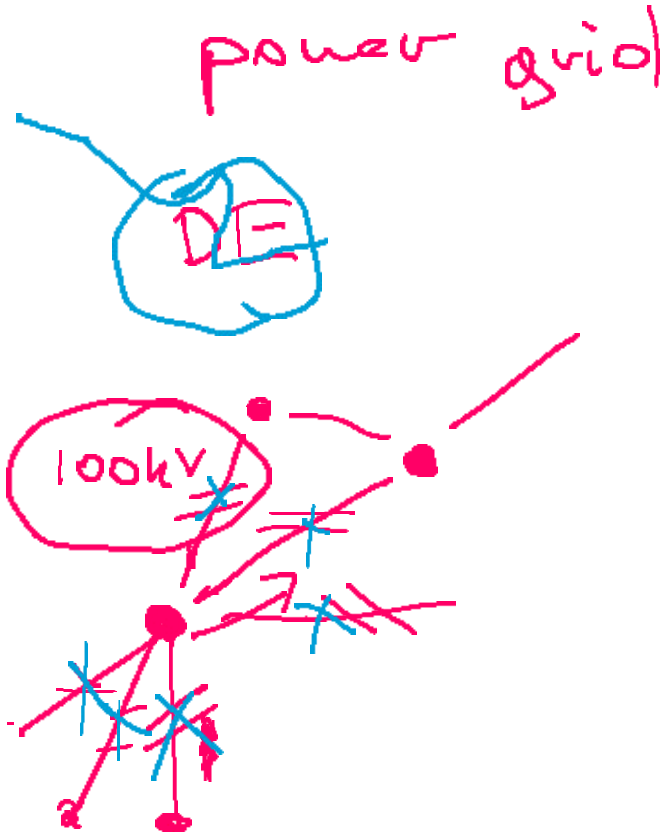
Types of errors:

--- failures of comm.

-- Byzantine communication

-- temporal subversion

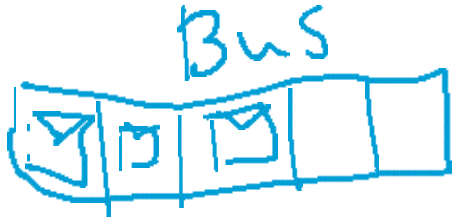
- local

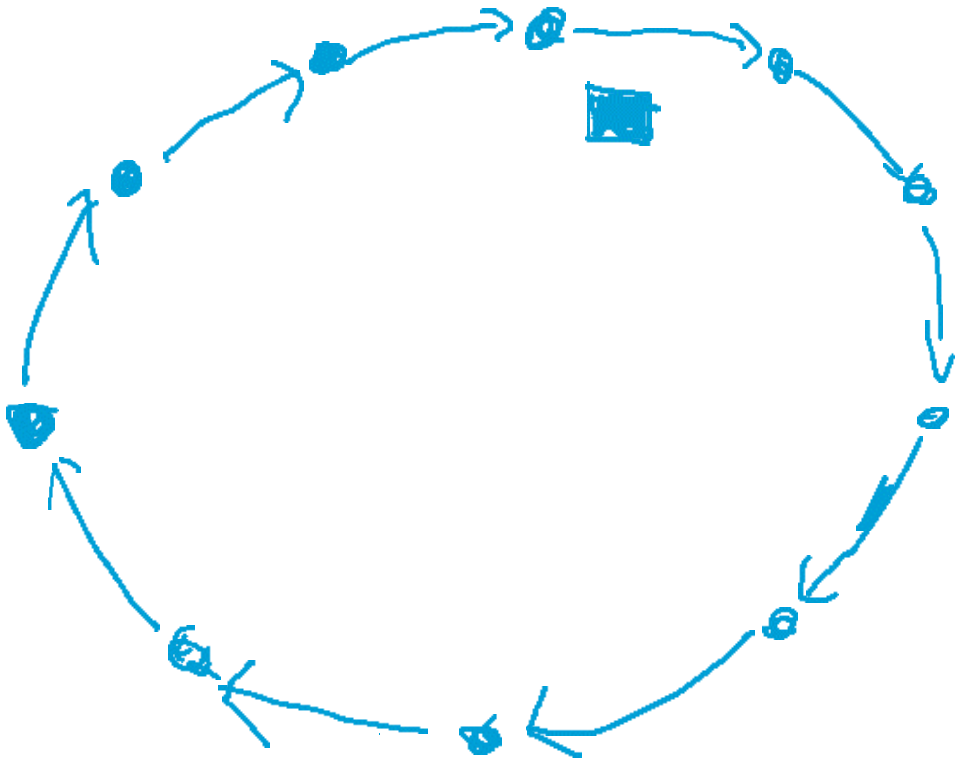


Self stabilization

End of problems transition to a proper shape

Token ring communication

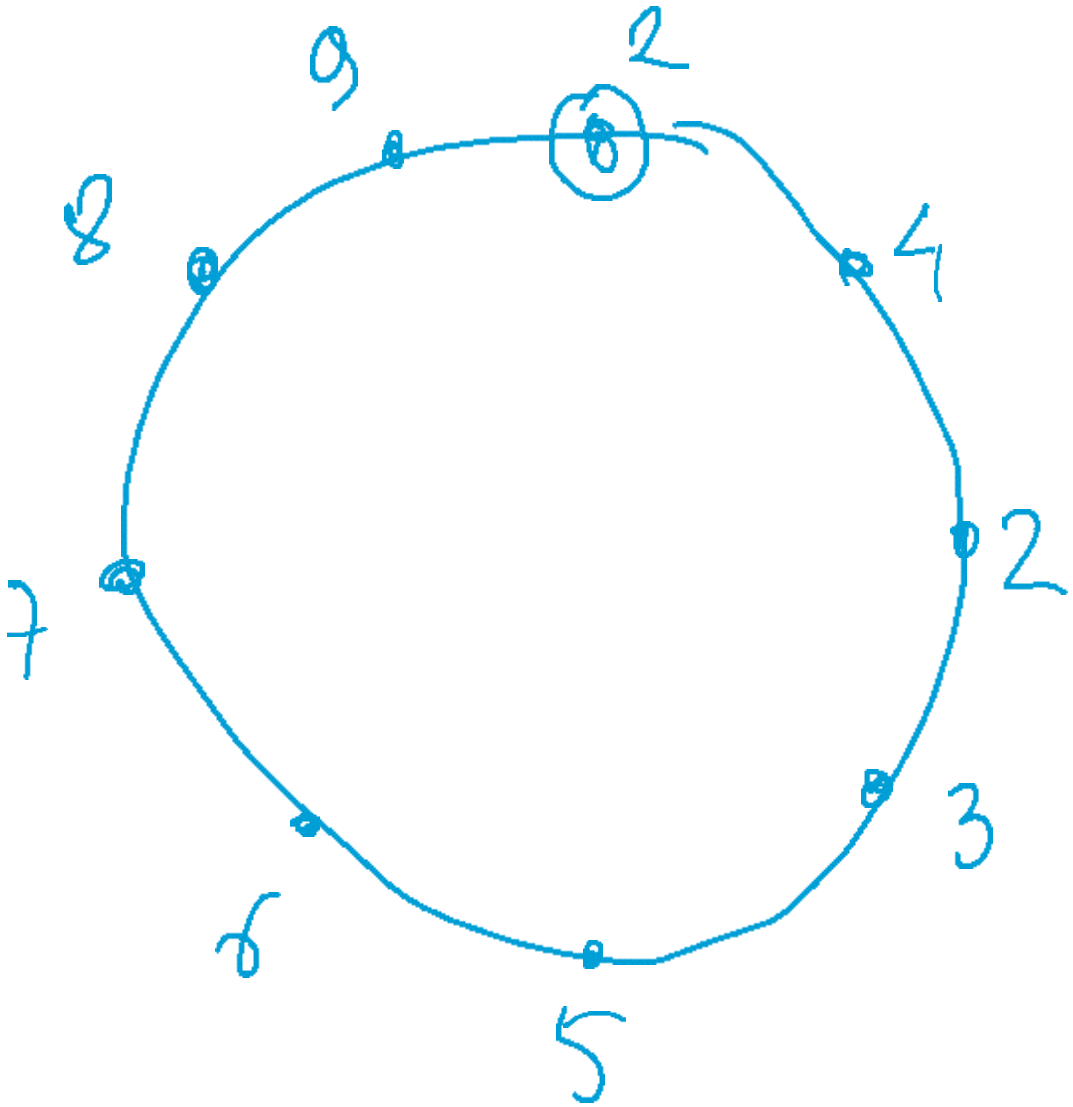




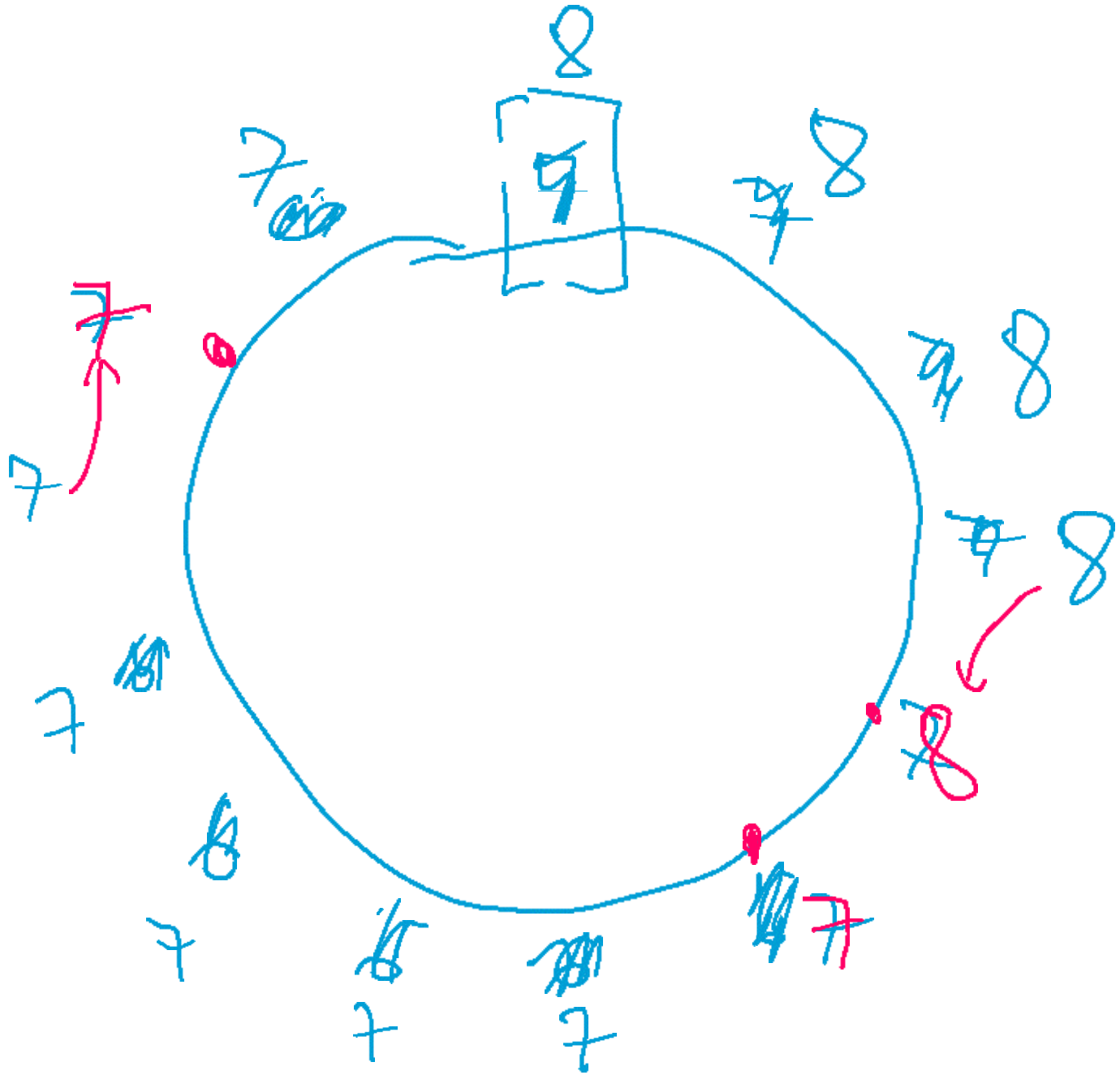
top p p(v)

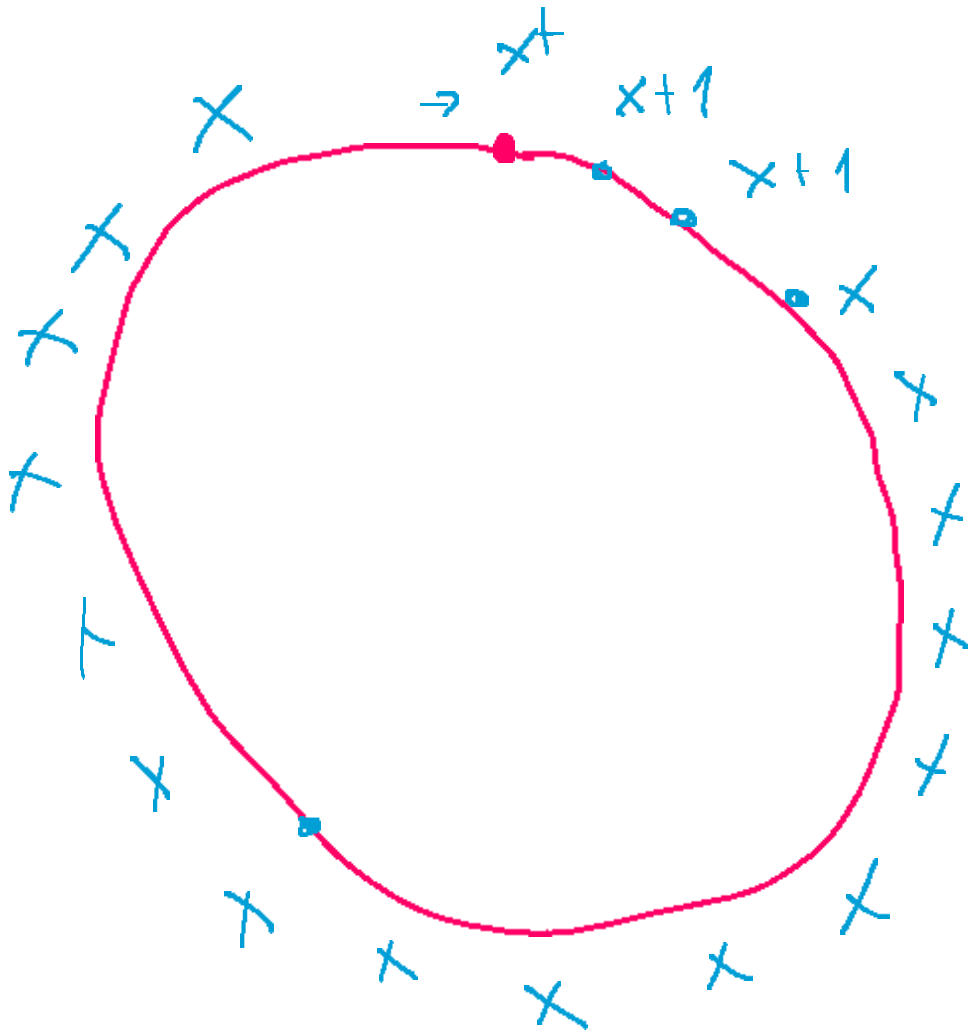
Algorithm 13.3 Self-stabilizing Token Ring

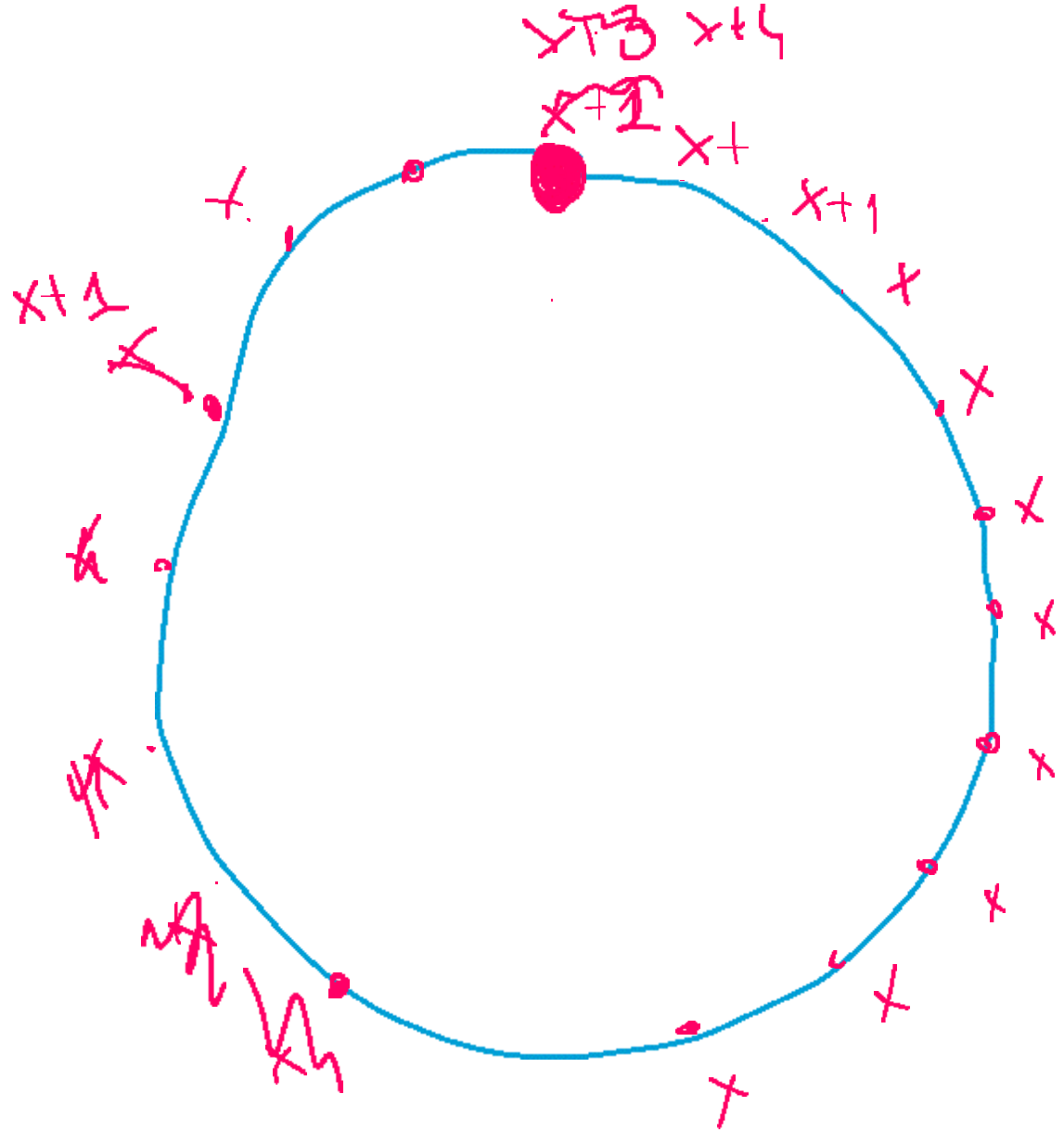
```
1: if v = v_0 then
2:     if S(v) = S(p) then
3:         S(v) := S(v) + 1 (mod n)
4:     end if
5: else
6:     S(v) := S(p)
7: end if
```



mess




x
x+1
x+1

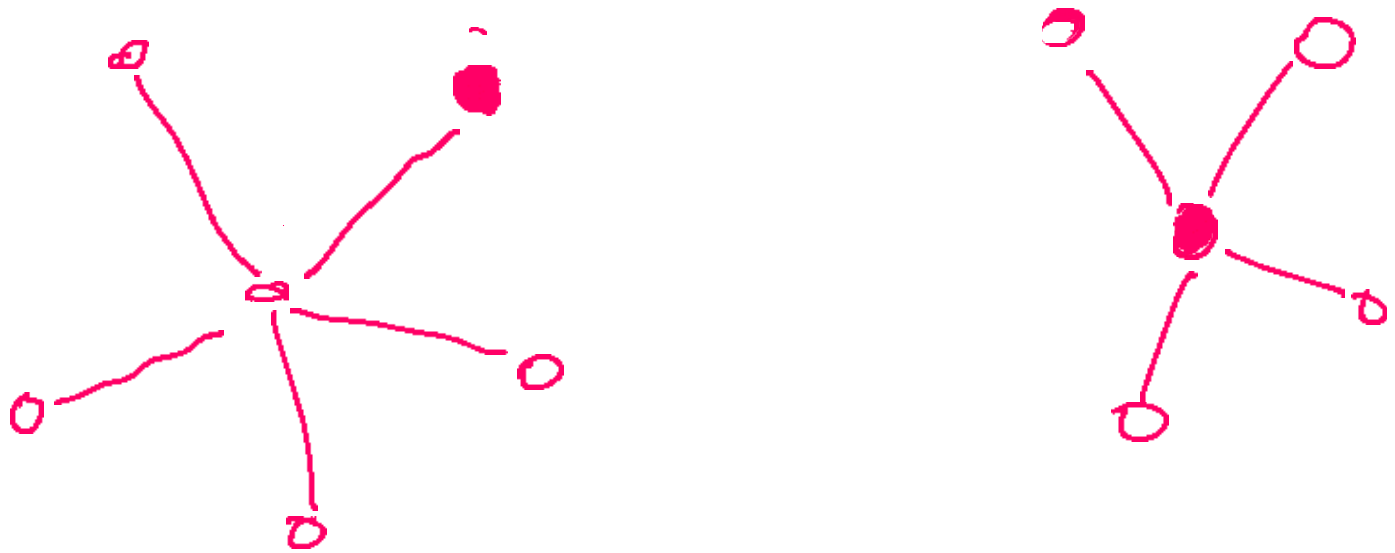


Algorithm 13.5 Self-stabilizing MIS

Require: Node IDs

Every node v executes the following code:

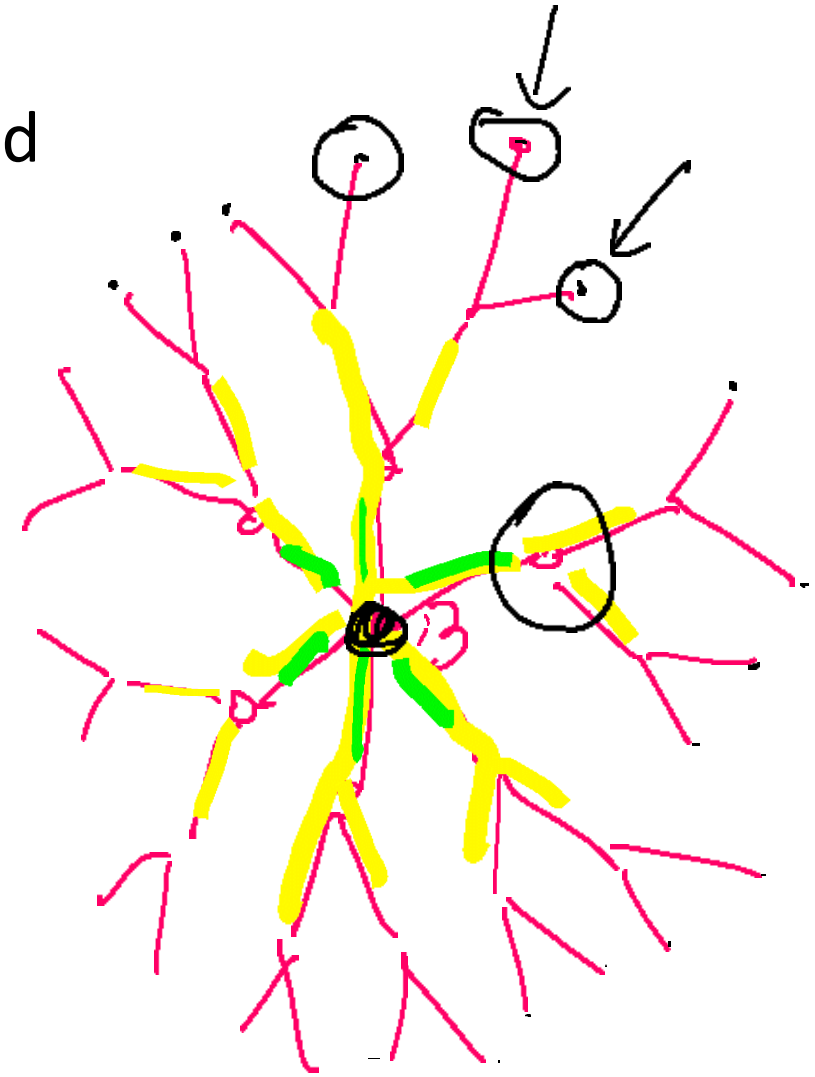
1: **do atomically**
2: Leave MIS if a neighbor with a larger ID is in the MIS
3: Join MIS if no neighbor with larger ID joins MIS
4: Send (node ID, MIS or not MIS) to all neighbors
5: **end do**



Transformation of deterministic algorithms into self-stabilization, time: d

idea: run d independent copies of the algorithm

neighborhood

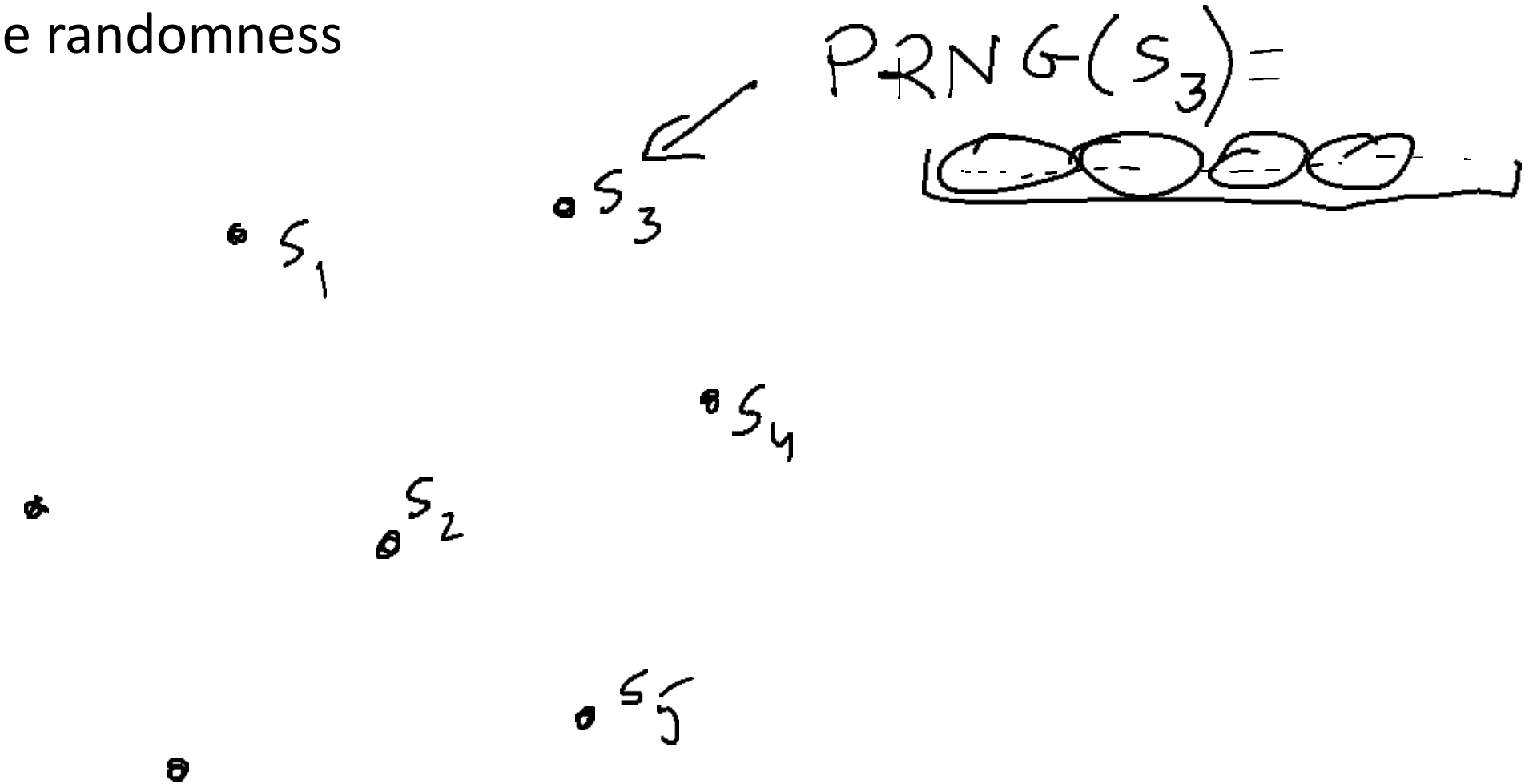


messages from radius b:

run algorithm — emulate

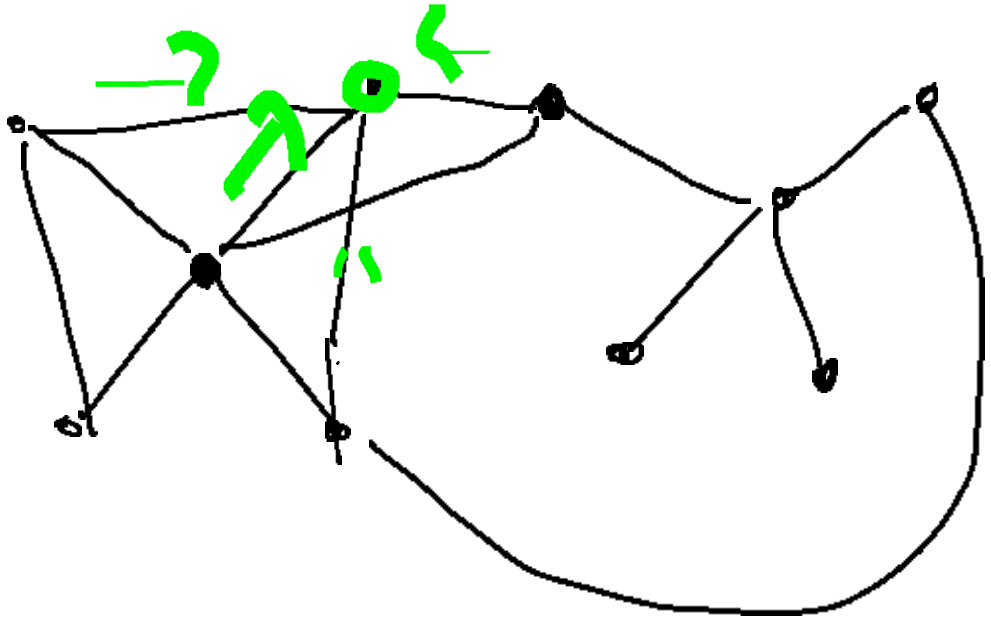
Randomized case

Verifiable randomness



Advance example of self-stabilization

Every evening:
-- a voter calls the friends
-- the friends give their recommendations
-- the voter changes preference according to majority of recommendations



- Is eventually everybody voting for the same party? .
- Will each citizen eventually stay with the same party? .
- Will citizens that stayed with the same party for some time, stay with that party forever? .
- And if their friends also constantly root for the same party? No.
- [illegible]

Theorem 13.7 (Dems & Reps). *Eventually every citizen is rooting for the same party every other day.*

1| Dep Dep Dep . . .

2| Rep De Re De Re . . .

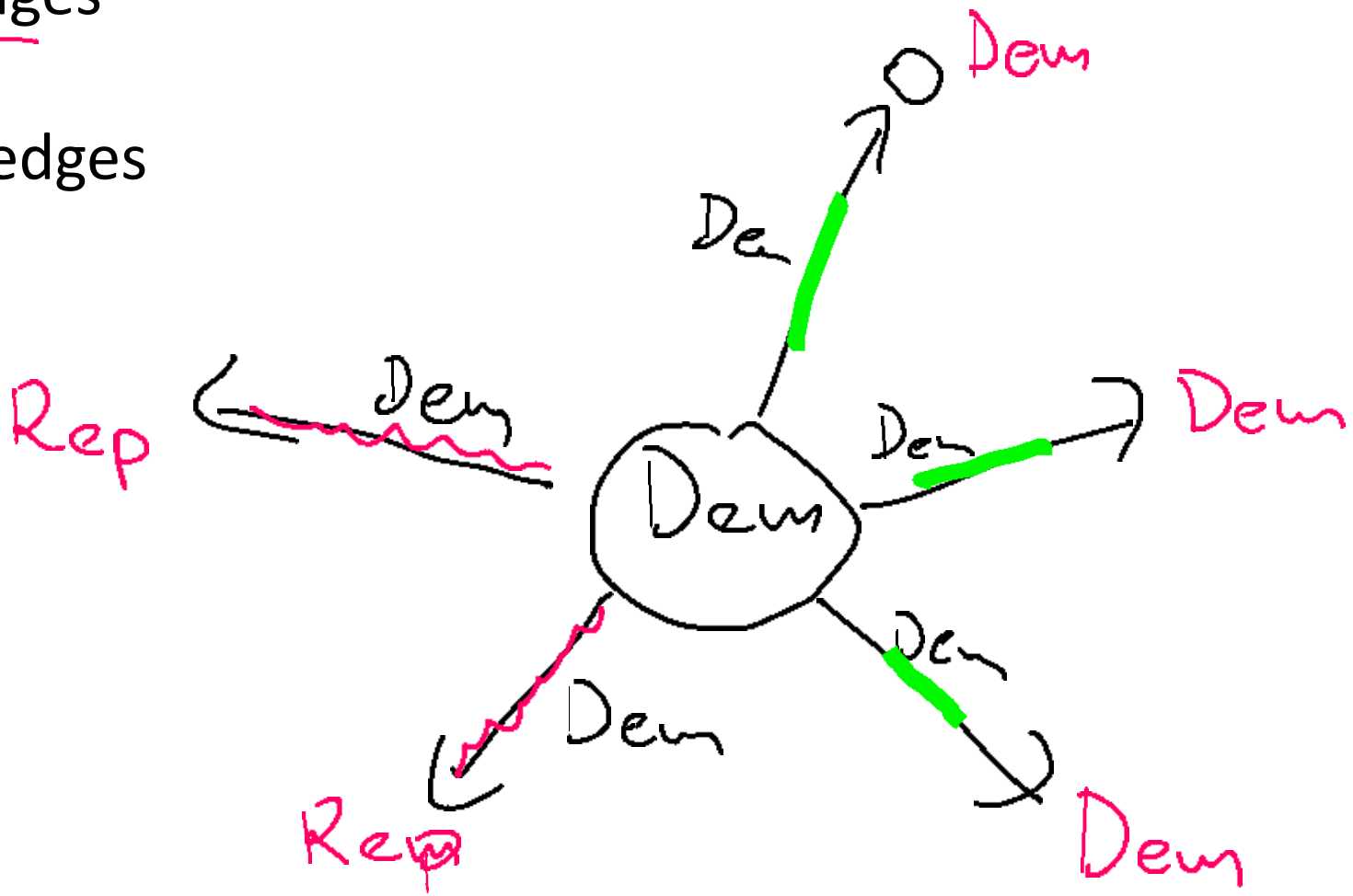
3| De Rep Re Rep . . .

4| Rep Rep Rep . . .

Day t: supporter of Dems

Bad out-edges

Good out-edges



Day t:

b bad out-edges from step t

g good out-edges from step t

Day t+1:

r recommendations for Reps

s recommendations for Dems

Case: $g > d$

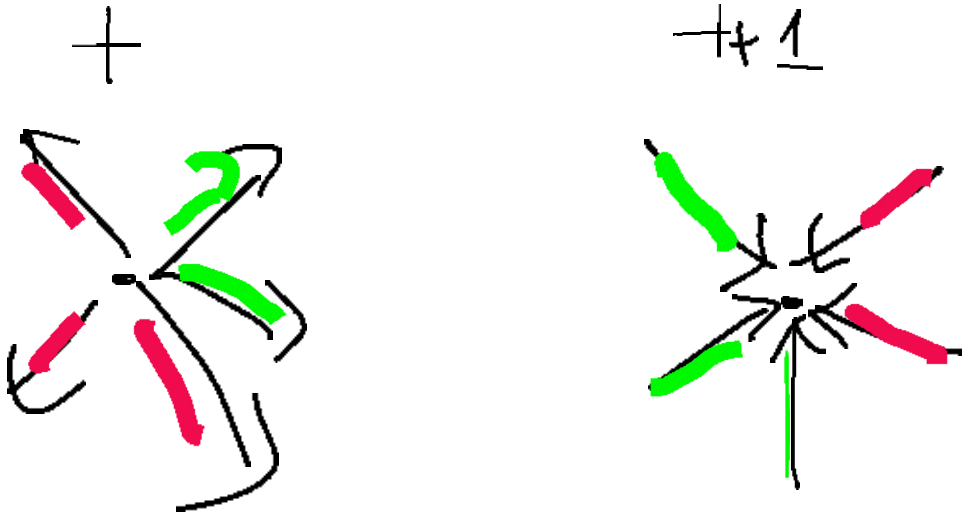
The voter roots Dems again

bad in-edges at step t+1 = # bad out-edges at step t

Case: g<b

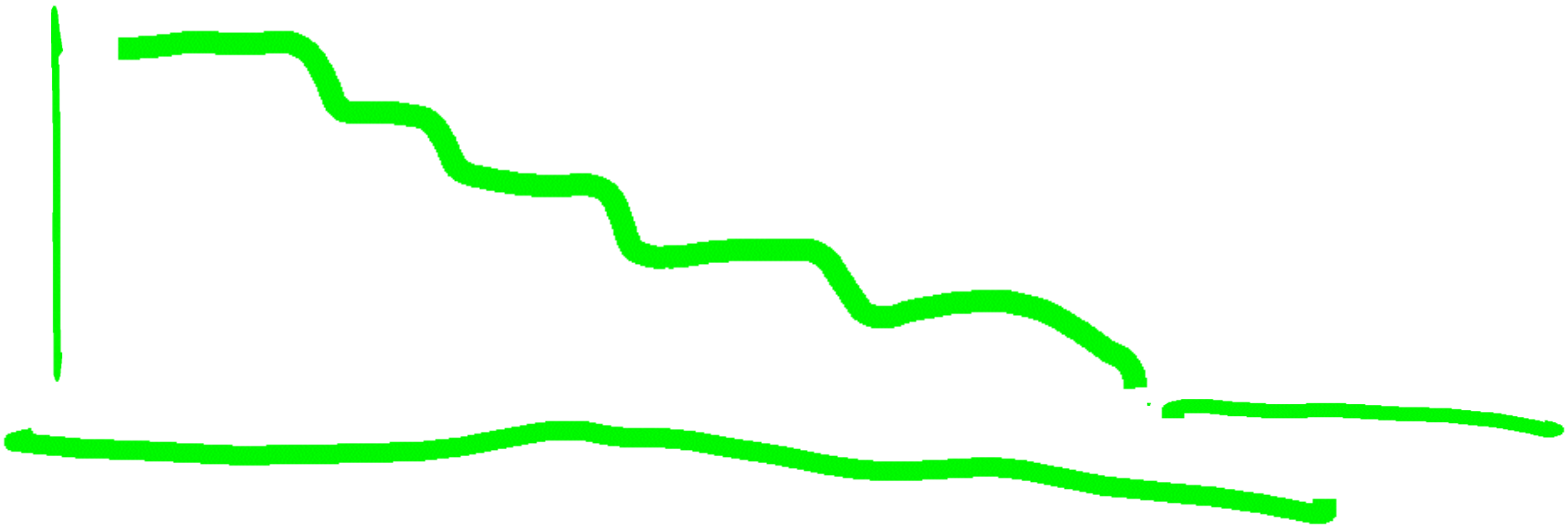
The voter roots REPs

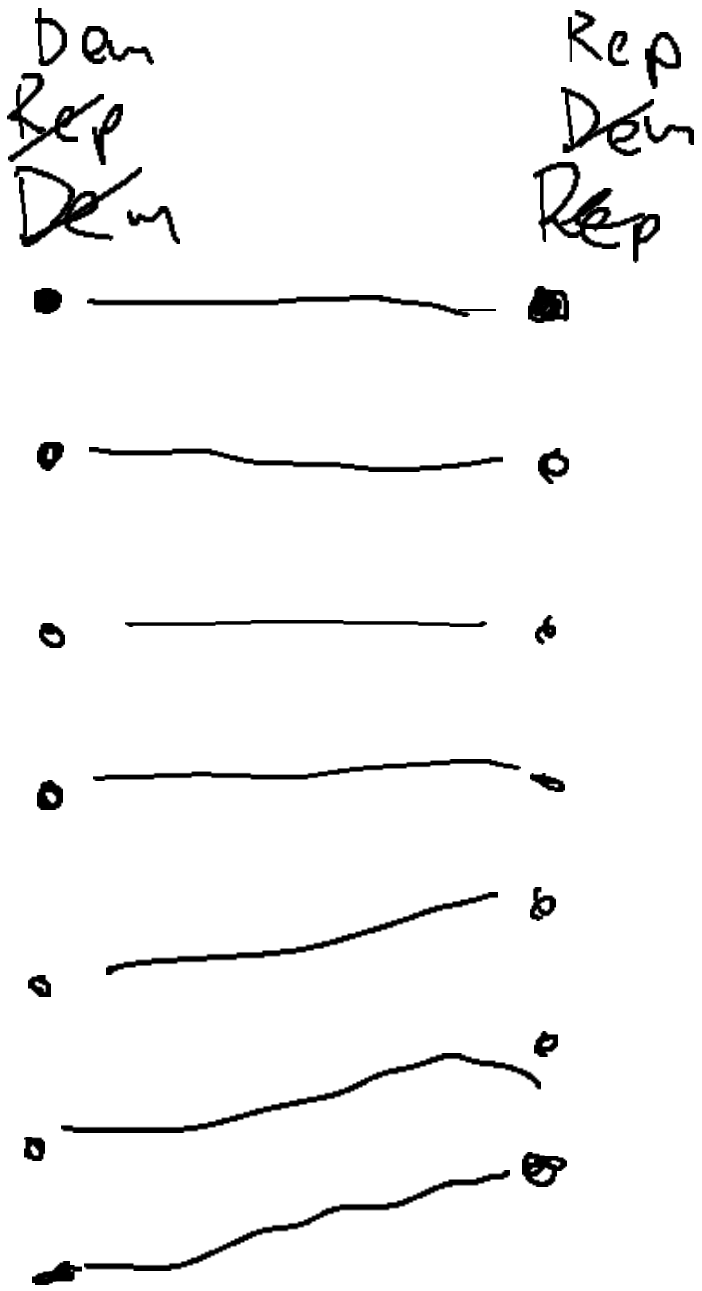
bad in-edges at step t+1 = g
bad out-edges at step t = b

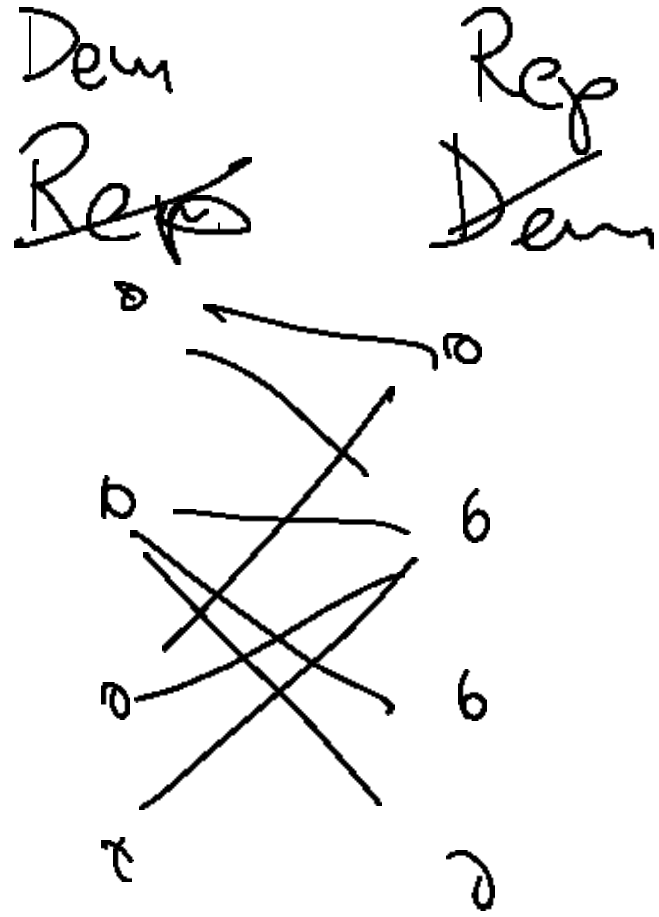


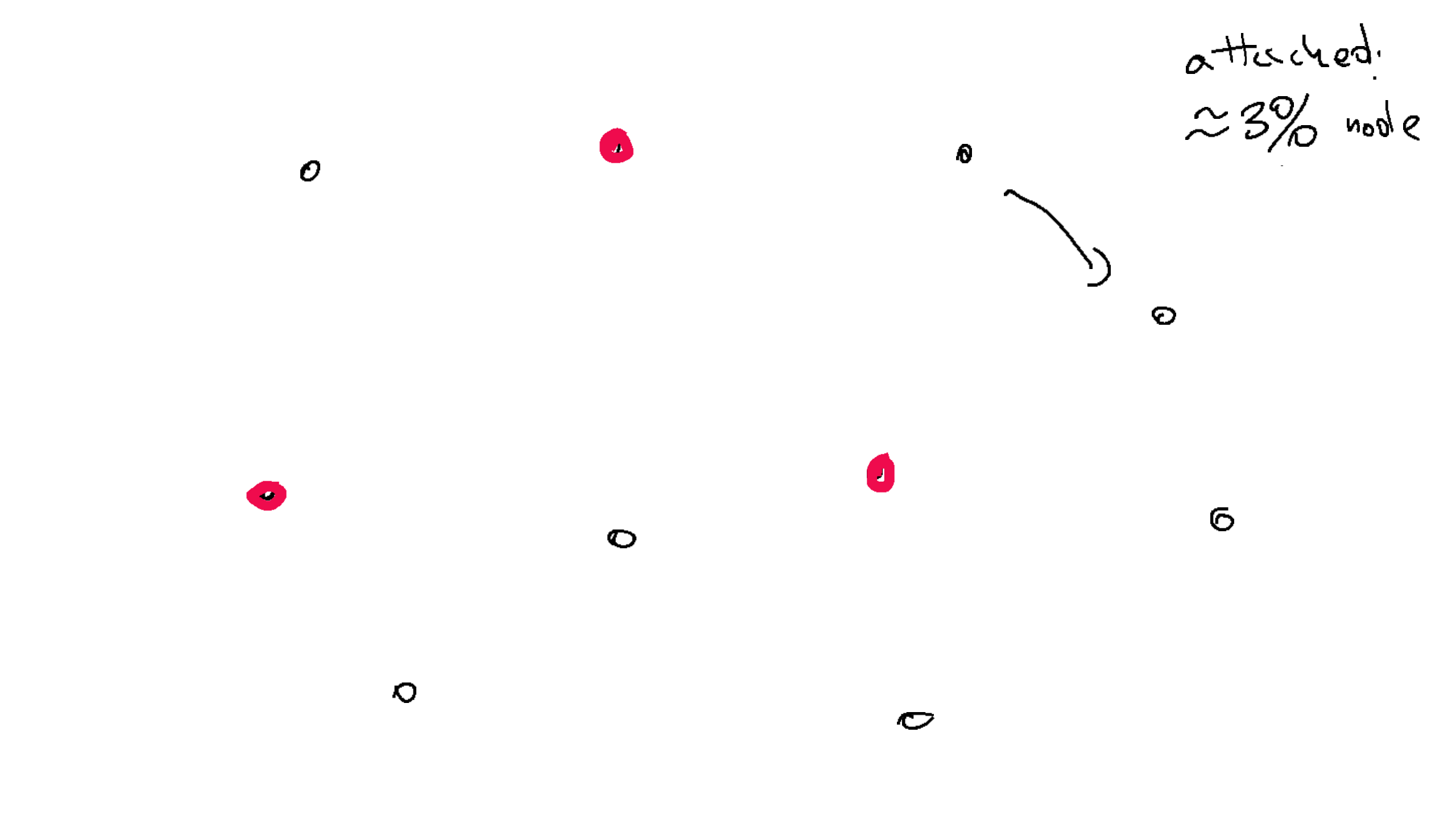
Total number of bad edges in the graph

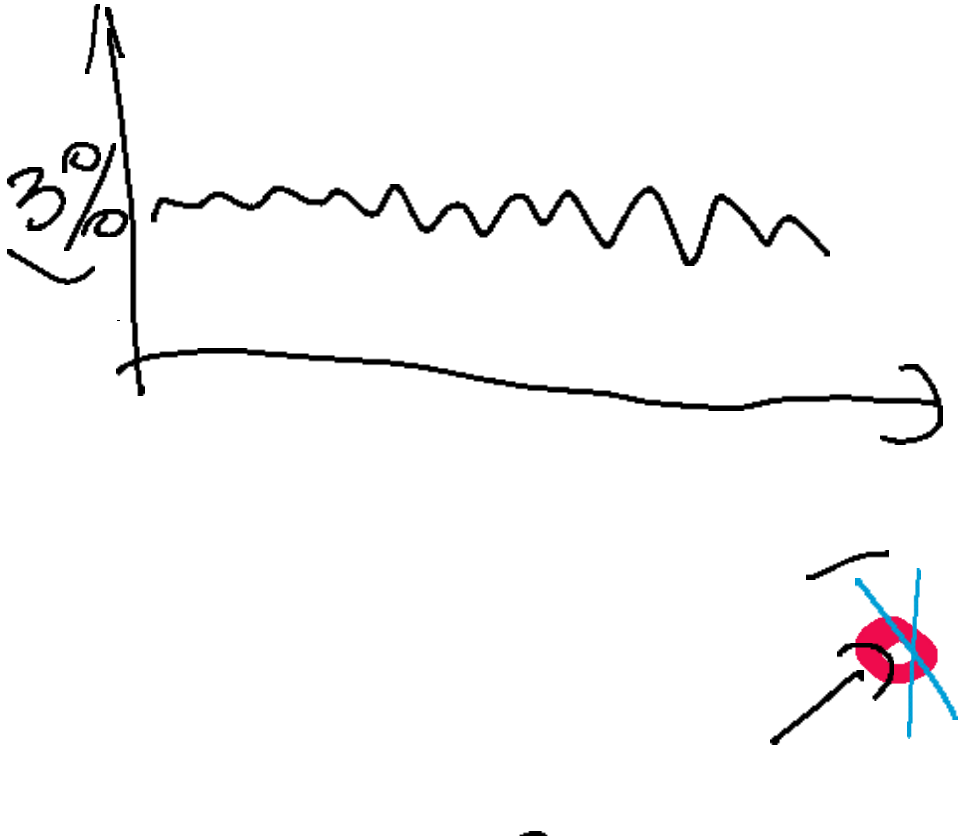
Total number of bad edges in the graph



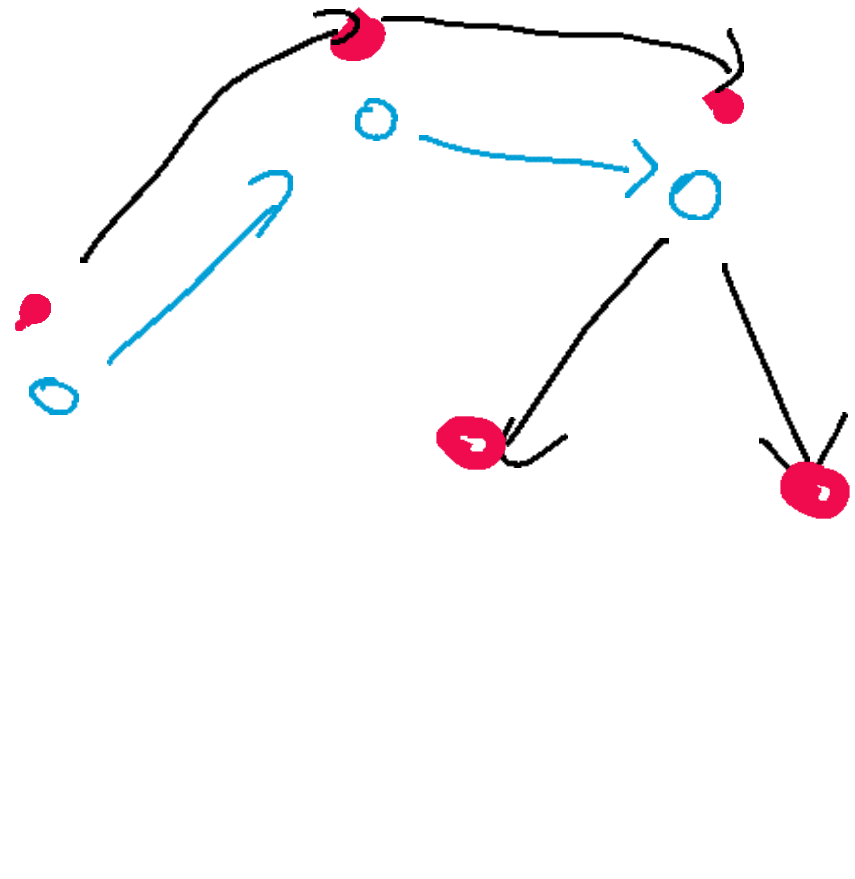

Dem
Rep
Rep
Dem


Dem
Rep
Rep
Dem


attacked:
≈ 3% node



a



b