Security and Game Theory

or

Out of Eden

Jared Saia

A hard fact:

Not everyone follows instructions

Good and Bad

A simple moral code for an aspiring deity (or computer scientist)

Good: follow instructions

Bad: don’t follow instructions

Question: How can we ensure that a group functions, even though some members of the group are bad?

Automata Studies
ed C. Shannon, 1956
Princ. Univ Press

PROBABILISTIC LOGICS AND THE SYNTHESIS OF RELIABLE ORGANISMS FROM UNRELIABLE COMPONENTS

J. von Neumann

1. INTRODUCTION

The paper that follows is based on notes taken by Dr. R. S. Pierce on five lectures given by the author at the California Institute of Technology in January 1952. They have been revised by the author but they reflect, apart from minor changes, the lectures as they were delivered.

The subject-matter, as the title suggests, is the role of error in logics, or in the physical implementation of logics — in automata-synthesis. Error is viewed, therefore, not as an extraneous and misdirected or misdirecting accident, but as an essential part of the process under consideration — its importance in the synthesis of automata being fully comparable to that of the factor which is normally considered, the intended and correct logical structure.

Components Fail, Group Functions

Group Decisions

- Periodically, components unite in a decision
- Idea: components vote. Problem: Who counts the votes?

Our Model

We assume an adversary controls a hidden subset of the components

We control the remaining components

Goal: All the good components unite in a decision

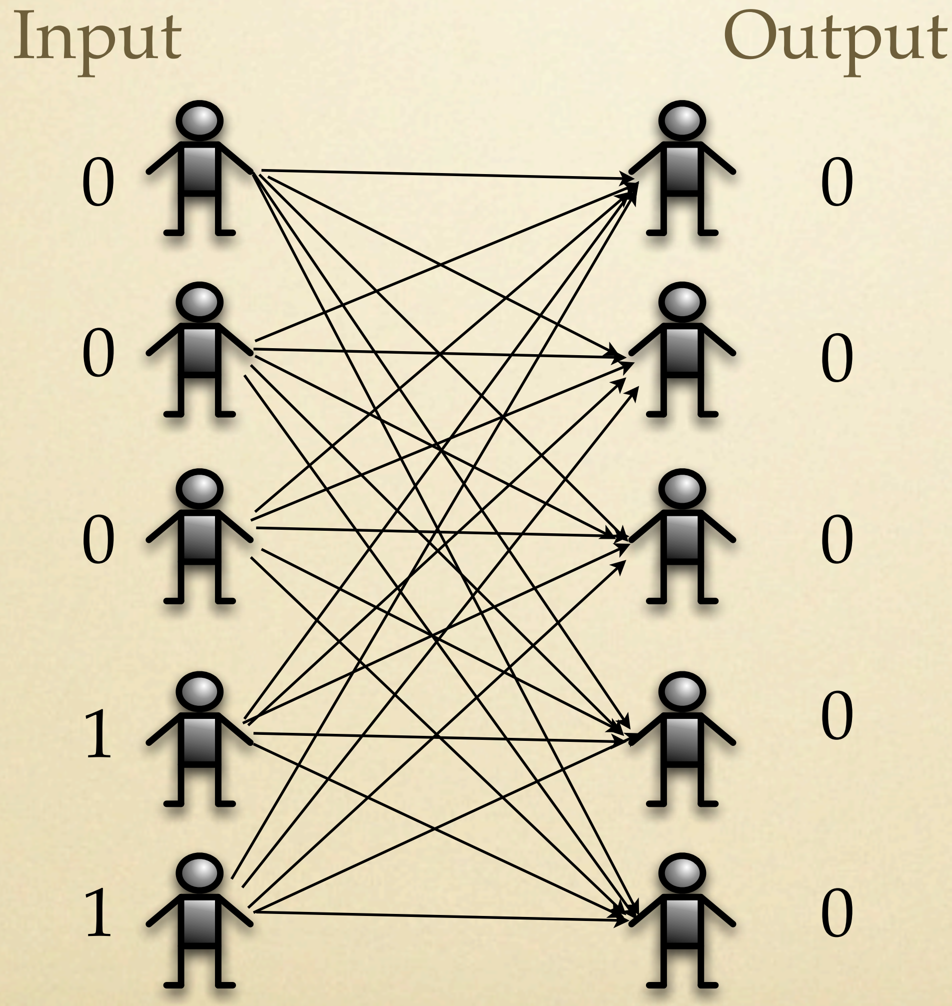
Our Model

We assume an adversary controls a hidden subset of the components

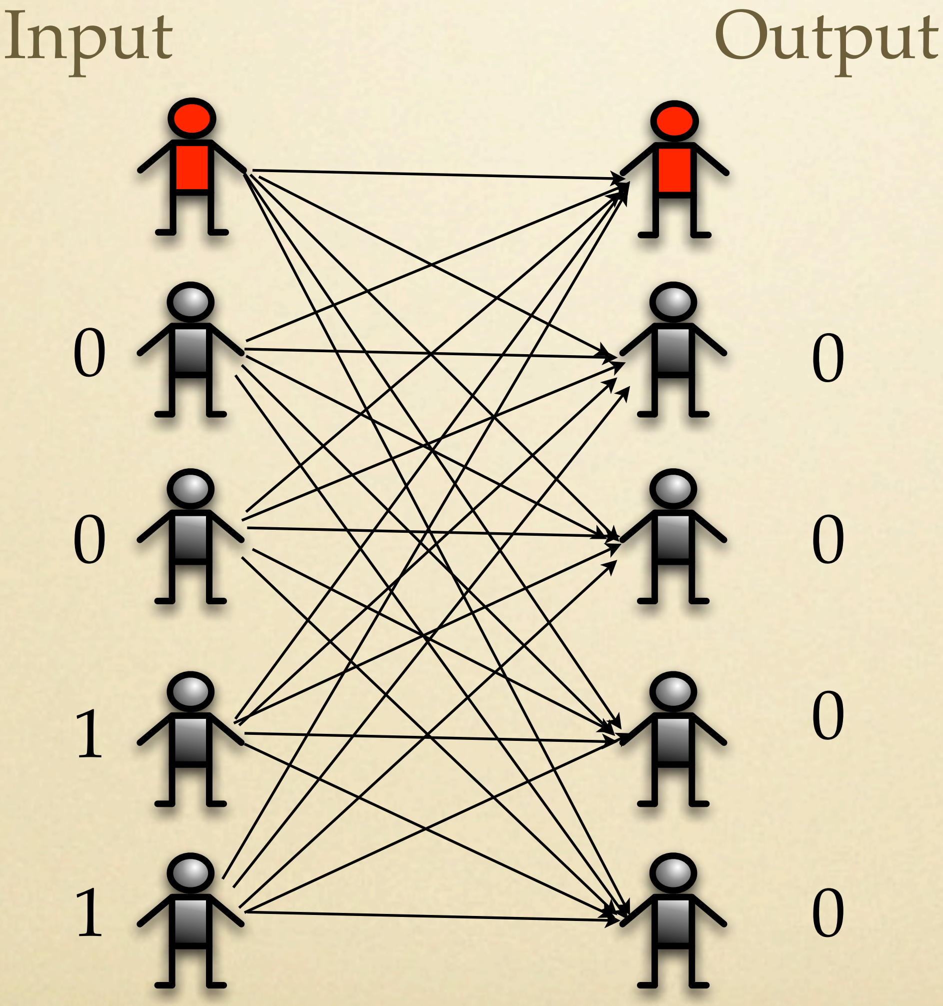
We control the remaining components

Goal: All the good components unite in a decision

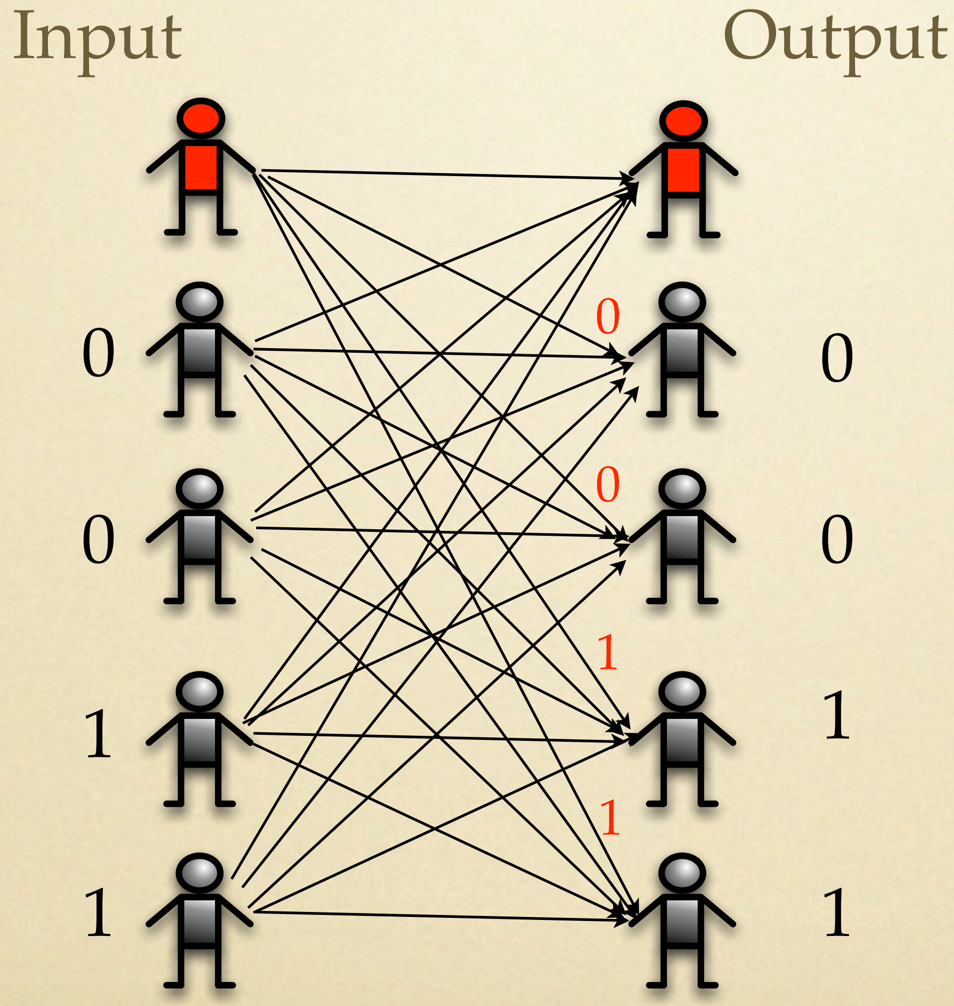
Idea: Majority Filtering



Idea: Majority Filtering



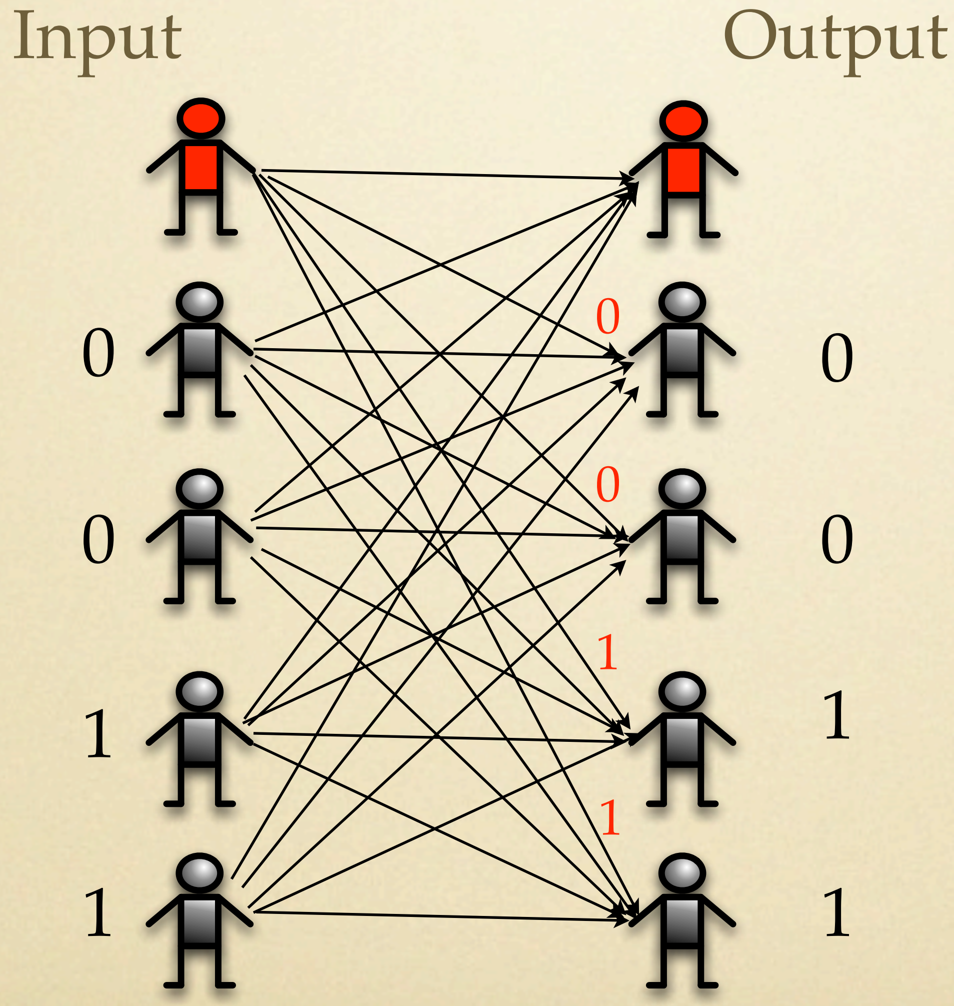
Problem



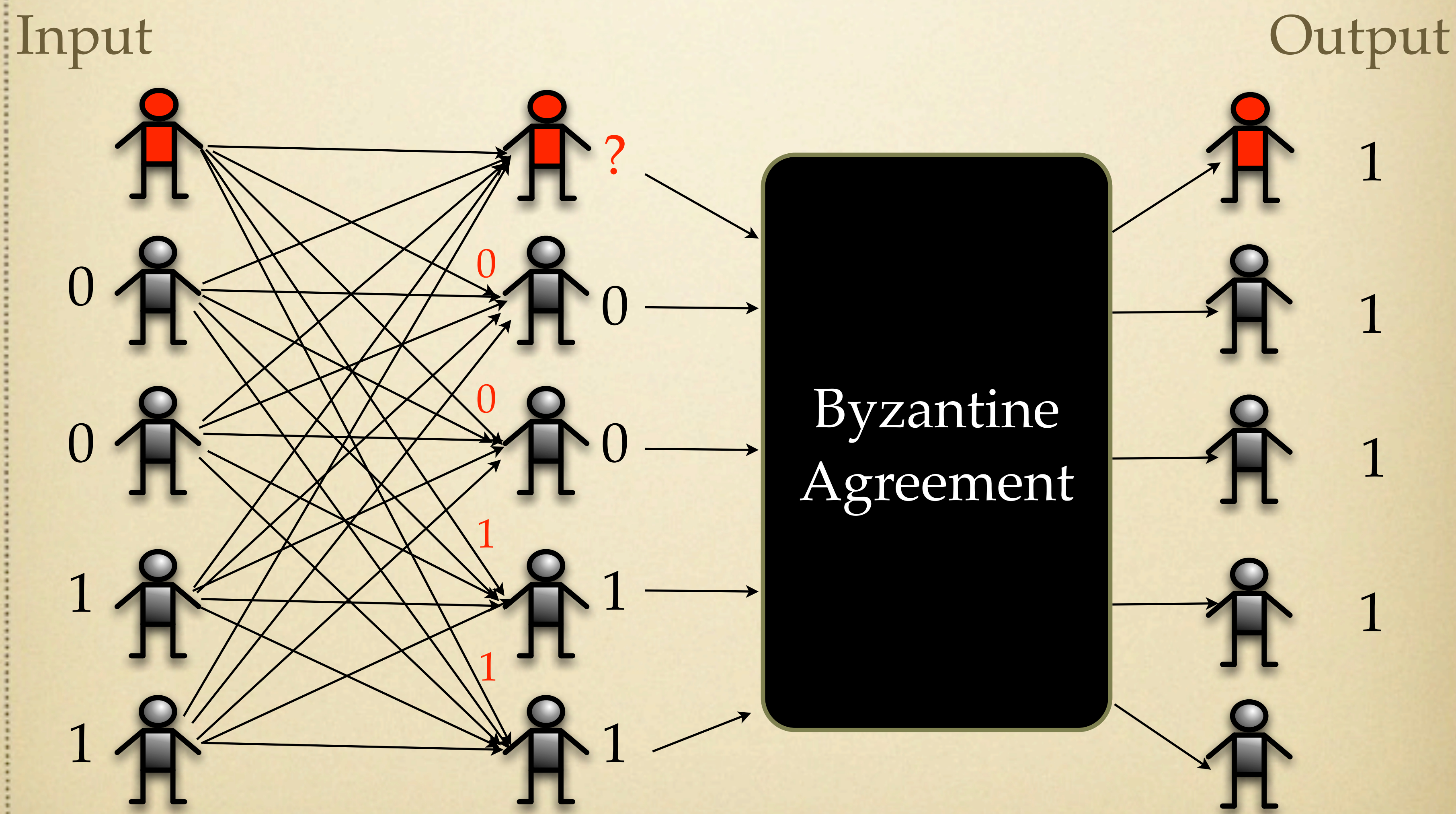
Byzantine Agreement

- Each processor starts with a bit
- Goal: 1) all good procs output the same bit; and 2) this bit equals an input bit of a good proc
- t = # bad procs controlled by an adversary

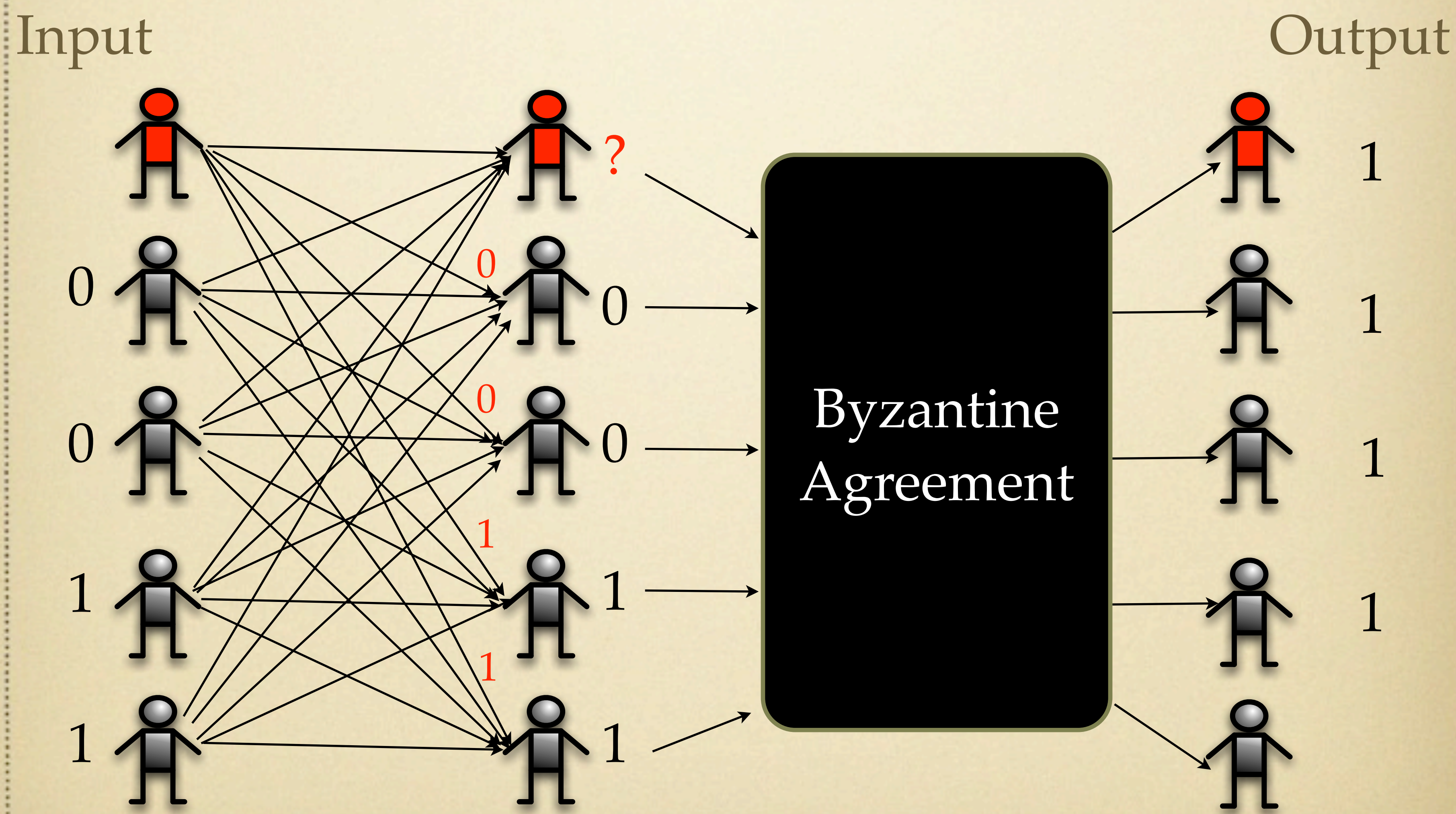
Problem



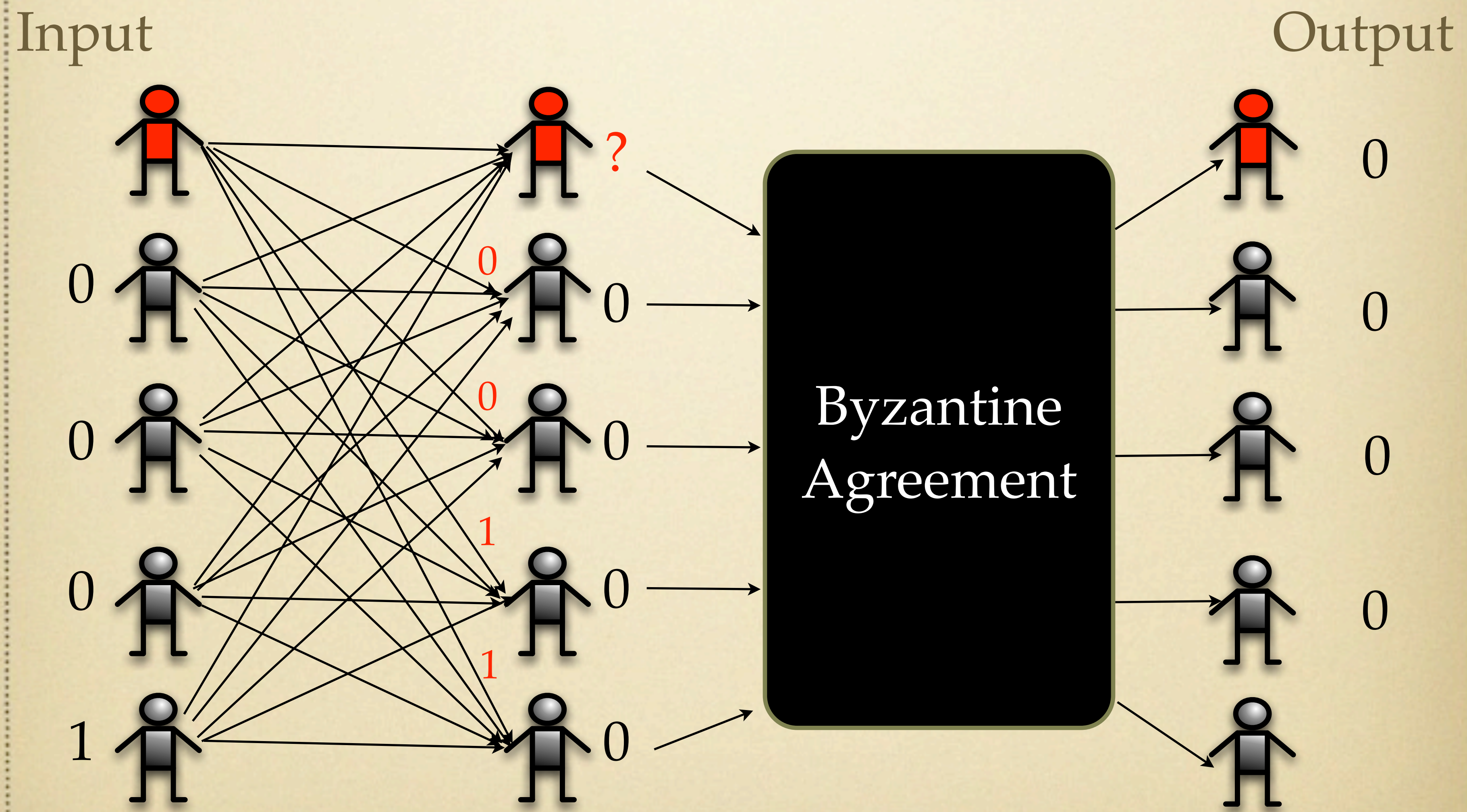
Idea



All good procs always output same bit



If majority bit held by >= 3 good procs, then all procs will output majority bit



Impossibility Result

- 1982: FLP show that 1 fault makes deterministic BA impossible
- 2007: Nancy Lynch wins Knuth Prize for this result, called "fundamental in all of Computer Science"

Applications

- Peer-to-peer networks

  *"These replicas cooperate with one another in a **Byzantine agreement** protocol to choose the final commit order for updates." [KBCCEGGRWWWZ '00]*

- Rule Enforcement

  *"... requiring the manager set to perform a **Byzantine agreement protocol"** [NWD '03]*

- Game Theory (Mediators)

  *"deep connections between implementing mediators and various agreement problems, such as **Byzantine agreement**" [ADH '08]*

Applications

- Peer-to-peer networks

  *"These replicas cooperate with one another in a **Byzantine agreement** protocol to choose the final commit order for updates." [KBCCEGGRWWWZ '00]*

- Rule Enforcement

  *"... requiring the manager set to perform a **Byzantine agreement protocol"** [NWD '03]*

- Game Theory (Mediators)

  *"deep connections between implementing mediators and various agreement problems, such as **Byzantine agreement**" [ADH '08]*

- Also: Databases, Sensor Networks, Cloud Computing, Control systems, etc.

Our Model

- Assume Global Coin: source of random bits that everyone can see
- Adversary: takes over 1/3 of the procs
- Private channels: message can be sent privately between any pair of procs

BA with Global Coin, GC

Rabin's Algorithm

Send your vote to everyone

Let *fraction* be fraction of votes for majority bit

If *fraction* >= 2/3, set vote to majority bit; else set vote to GC

BA with Global Coin, GC

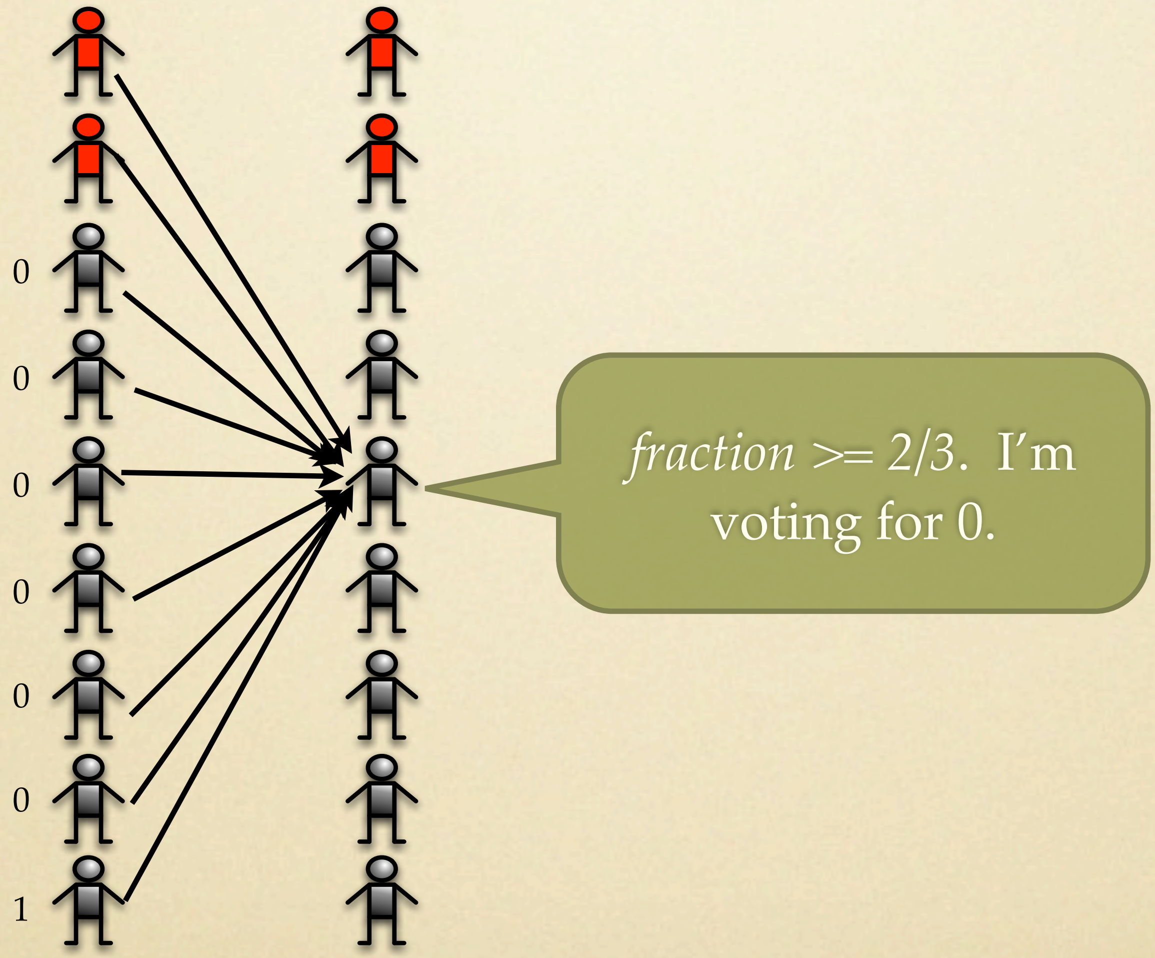
Rabin's Algorithm

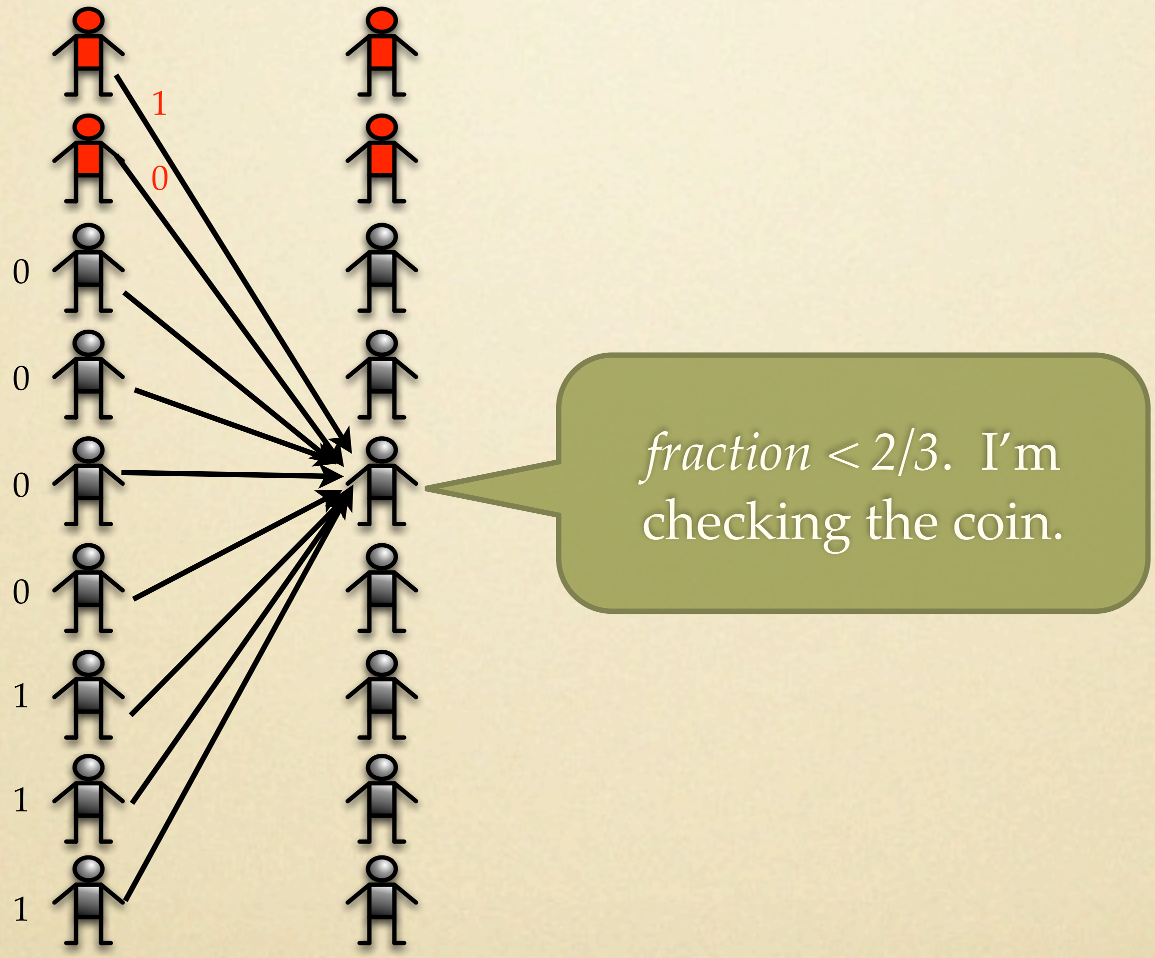
Set your vote to input bit

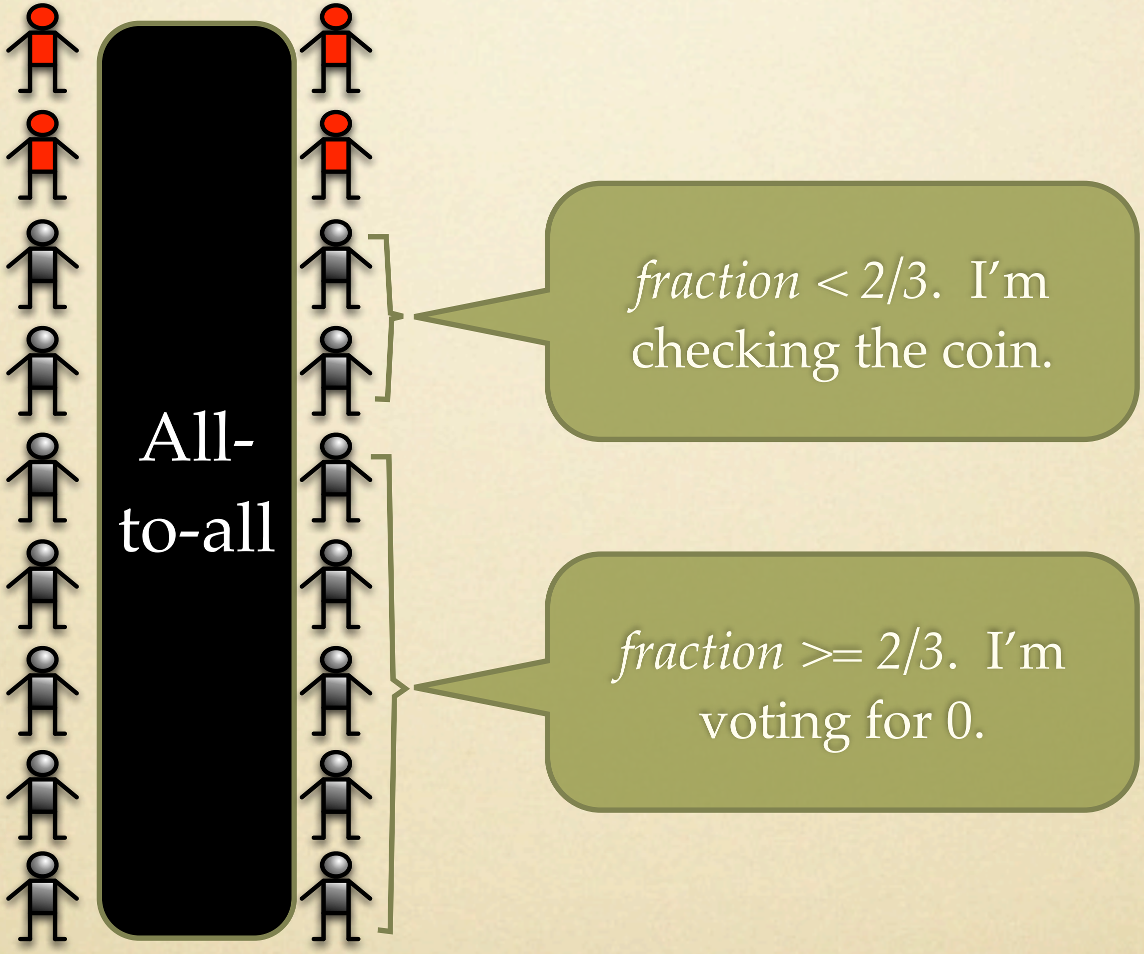
Repeat clogn times:

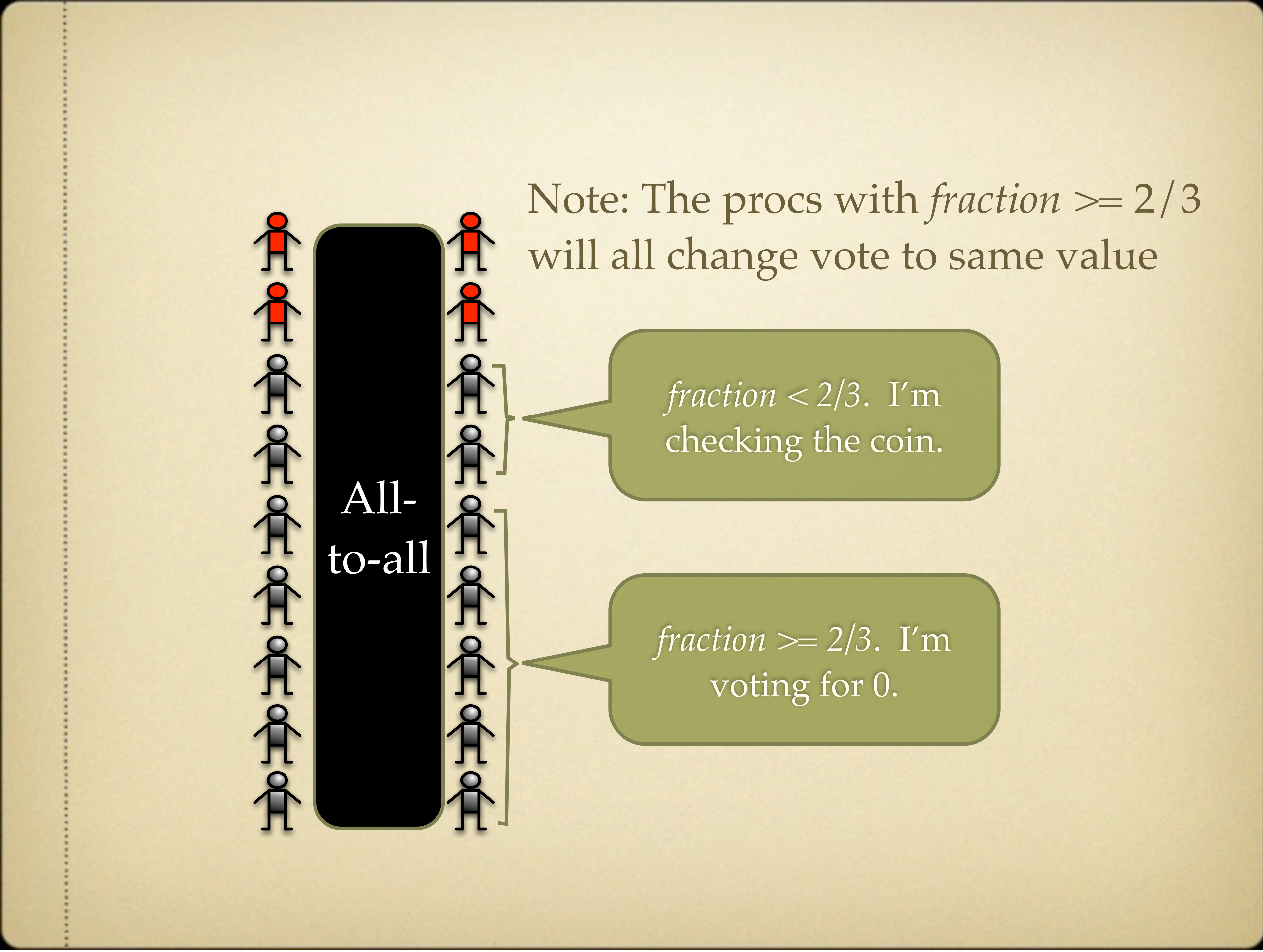
- Send your vote to everyone
- Let *fraction* be fraction of votes for majority bit
- If *fraction* >= 2/3, set vote to majority bit; else set vote to GC

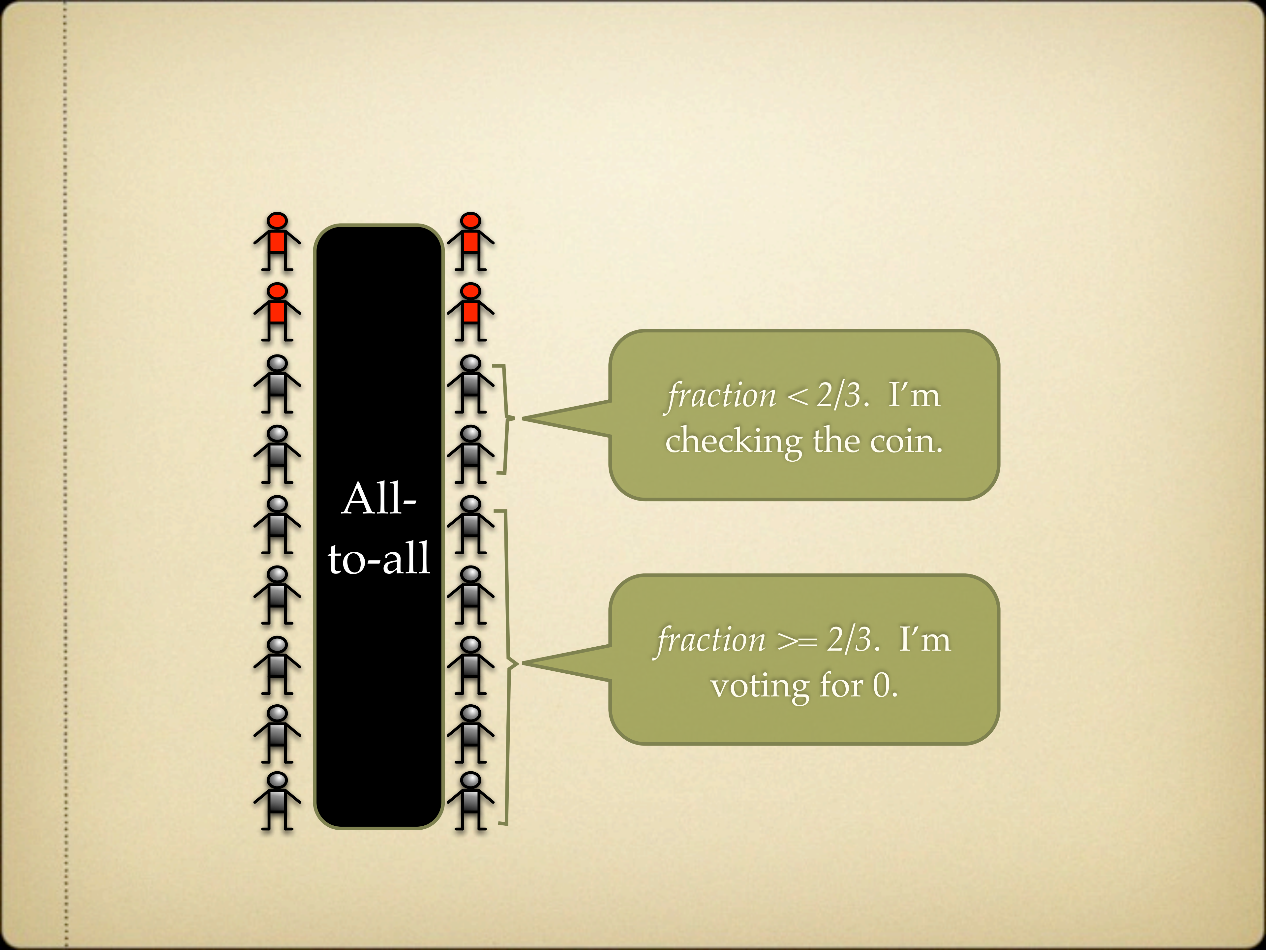
Output your vote


0
0
0
0
0
0
1
fraction >= 2/3. I'm voting for 0.

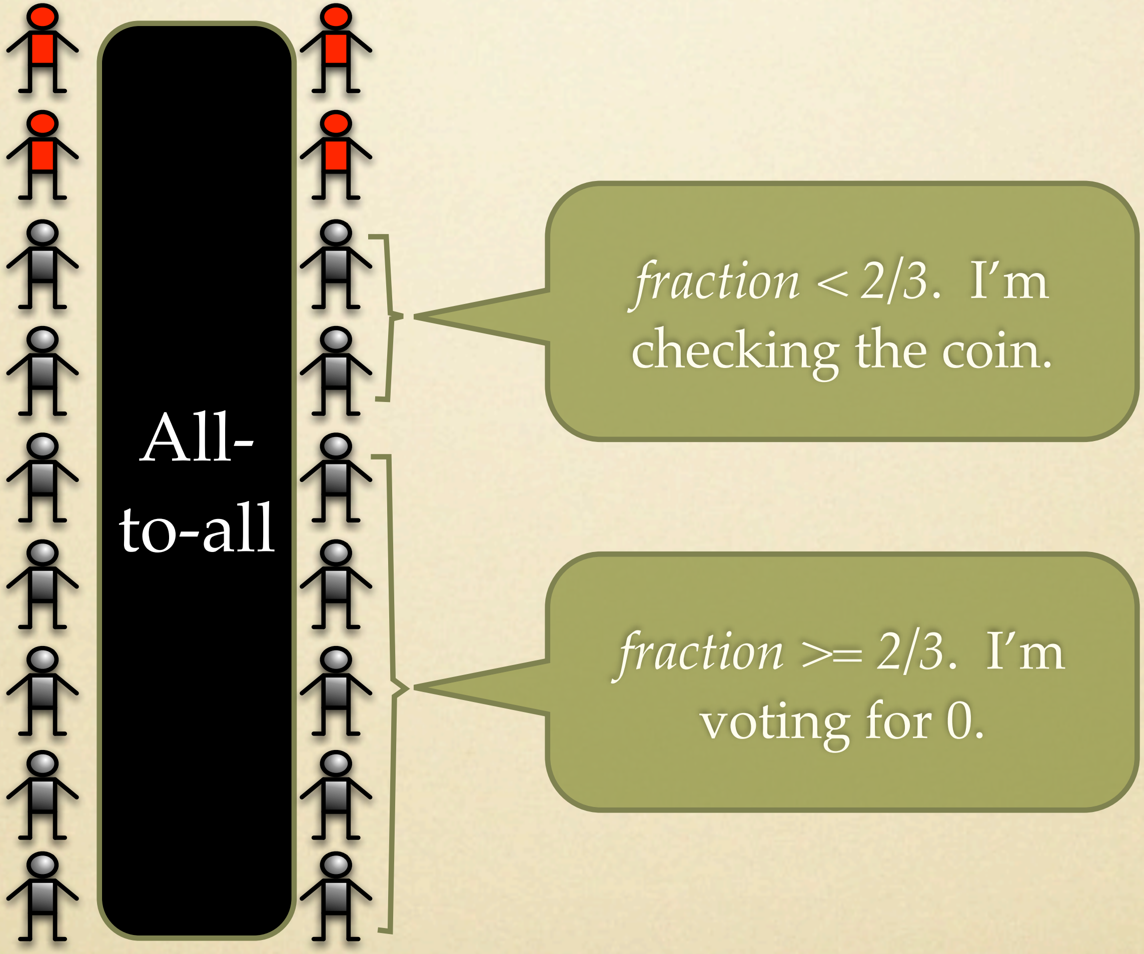

1
0
0
0
0
0
1
1
1
fraction < 2/3. I'm checking the coin.


All-to-all
fraction < 2/3. I'm checking the coin.
fraction >= 2/3. I'm voting for 0.


Note: The procs with *fraction* >= 2/3
will all change vote to same value
All-
to-all
fraction < 2/3. I'm
checking the coin.
fraction >= 2/3. I'm
voting for 0.

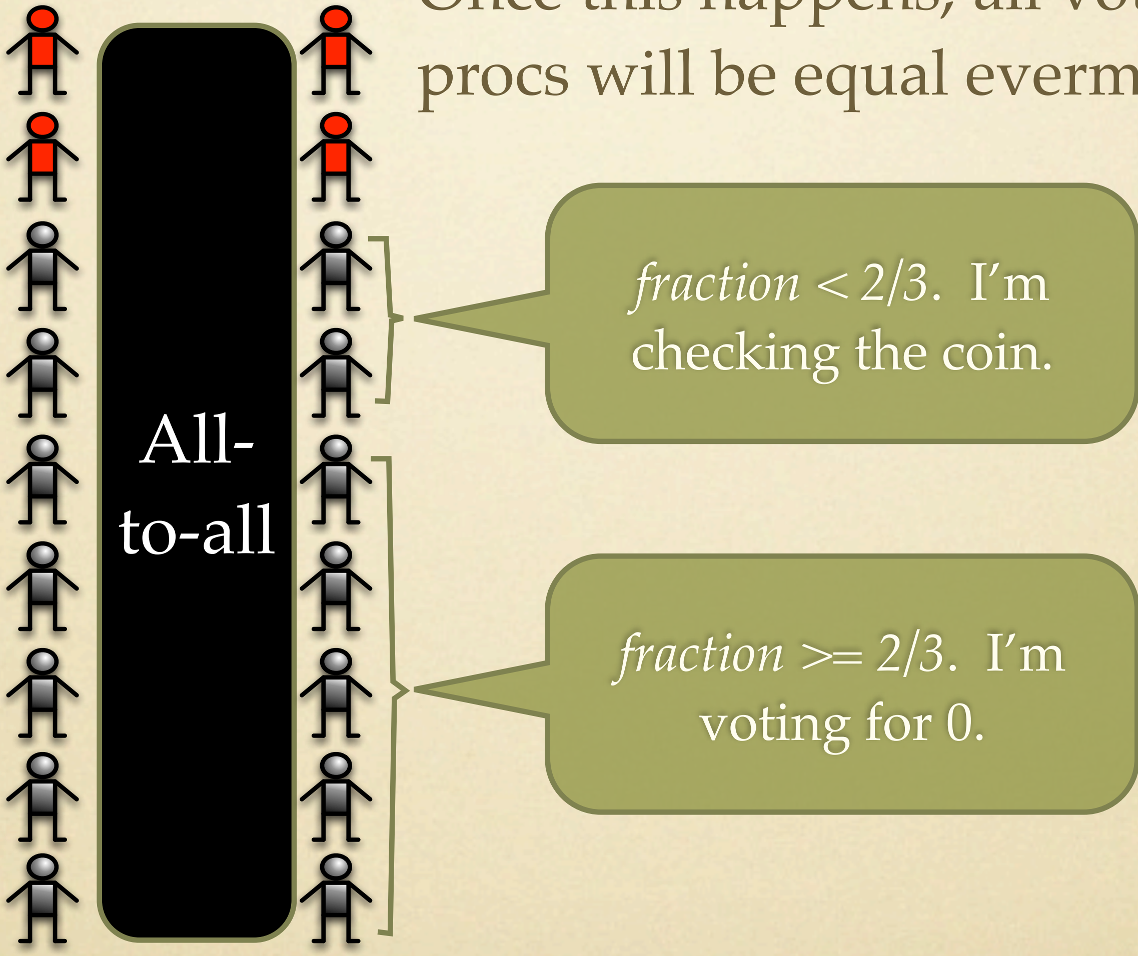

All-
to-all
fraction < 2/3. I'm
checking the coin.
fraction >= 2/3. I'm
voting for 0.

Probability 1/2 that both groups change vote to the same value



Probability 1/2 that both groups change vote to the same value

Once this happens, all votes of good procs will be equal evermore



Probability 1/2 that both groups change vote to the same value

Once this happens, all votes of good procs will be equal evermore

$$\text{Prob of failure} = (1/2)^{c \log n}$$
$$= 1/n^c$$

Probability 1/2 that both groups change vote to the same value

Once this happens, all votes of good procs will be equal evermore

$$\text{Prob of failure} = (1/2)^{clogn}$$
$$= 1/n^c$$

$$\text{Prob of success} = 1 - 1/n^c$$

Probability 1/2 that both groups change vote to the same value

Once this happens, all votes of good procs will be equal evermore

$$\text{Prob of failure} = (1/2)^{c \log n}$$
$$= 1/n^c$$

$$\text{Prob of success} = 1 - 1/n^c$$

whp

Global Coin

- Q: Where can we get a global coin?
- A1: The procs take turns flipping a coin and sending the results to everyone. The good procs at least will flip a fair coin.
- Problem: If n procs, this method may take ~n rounds
- A2: Parity of closing price of stock market

Leader Election

- n processors
- Less than a 1/3 fraction of them are bad
- Goal: Elect a leader such that 1) all good procs agree on the leader; and 2) the leader has constant probability of being good

Committee Election

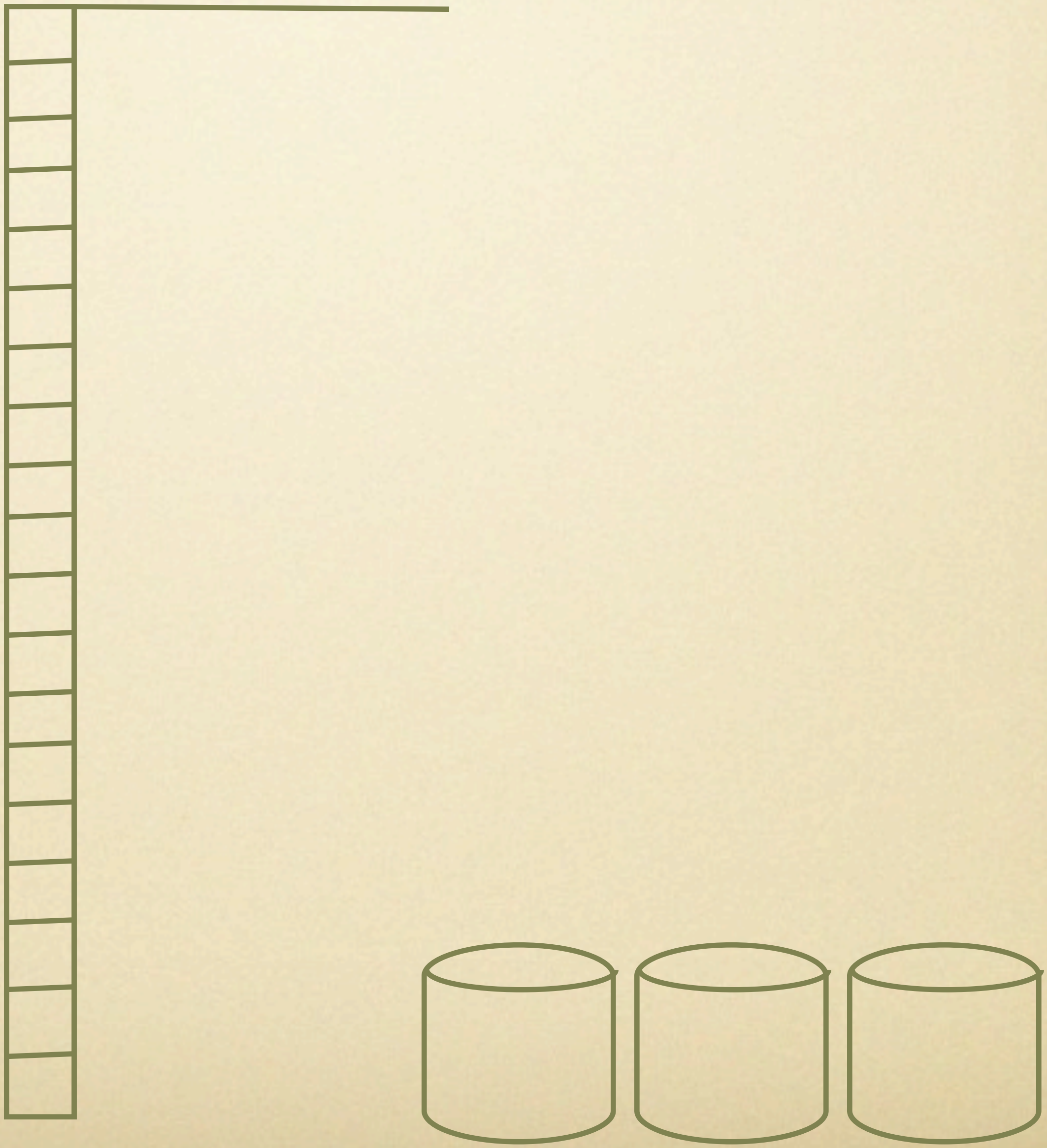
- n processors
- Less than a 1/3 fraction of them are bad
- Goal: Elect a committee such that 1) all good procs agree on the committee; and 2) the fraction of bad procs in the committee isn't too large

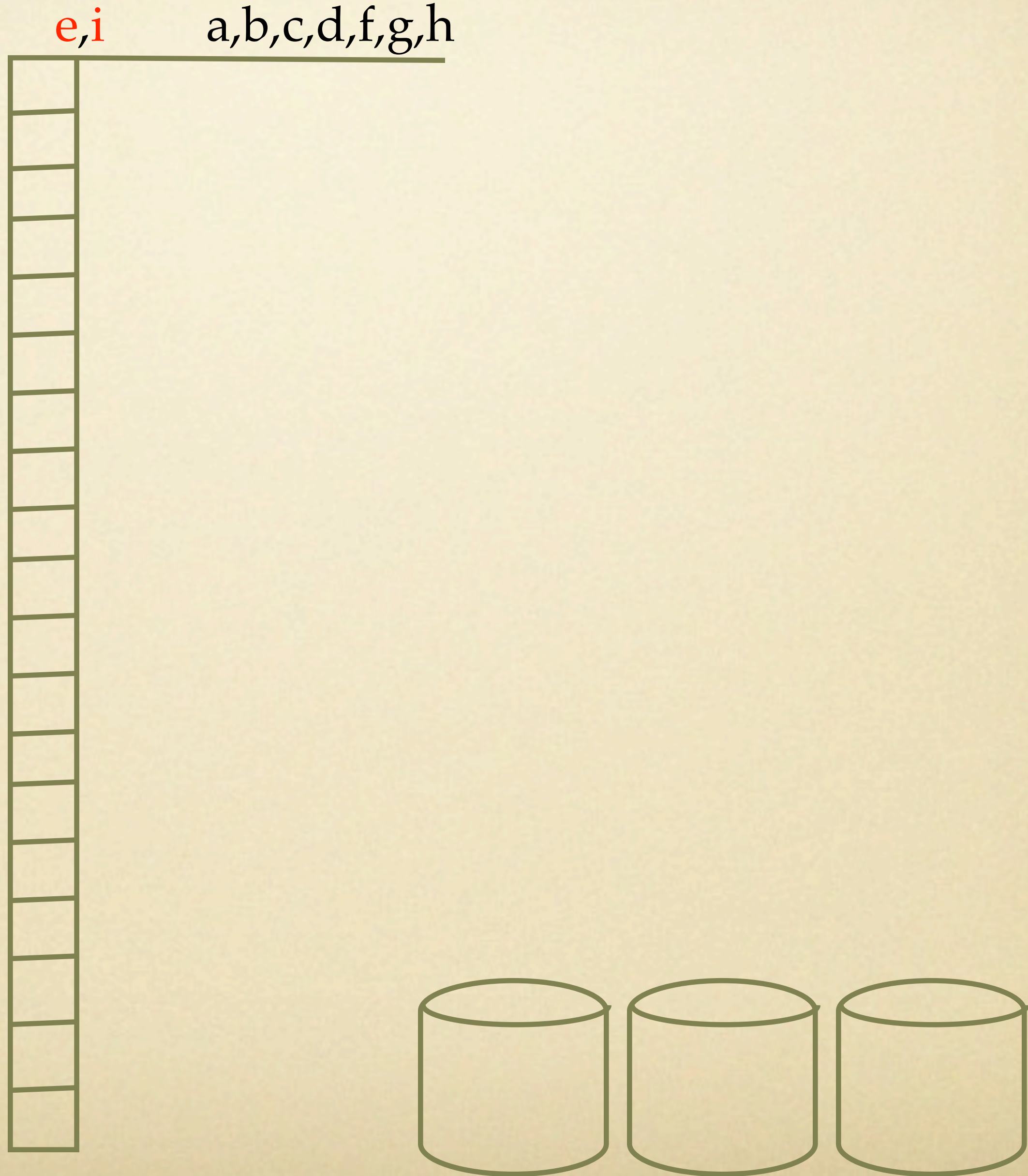
Idea: Lightest Bin Algorithm

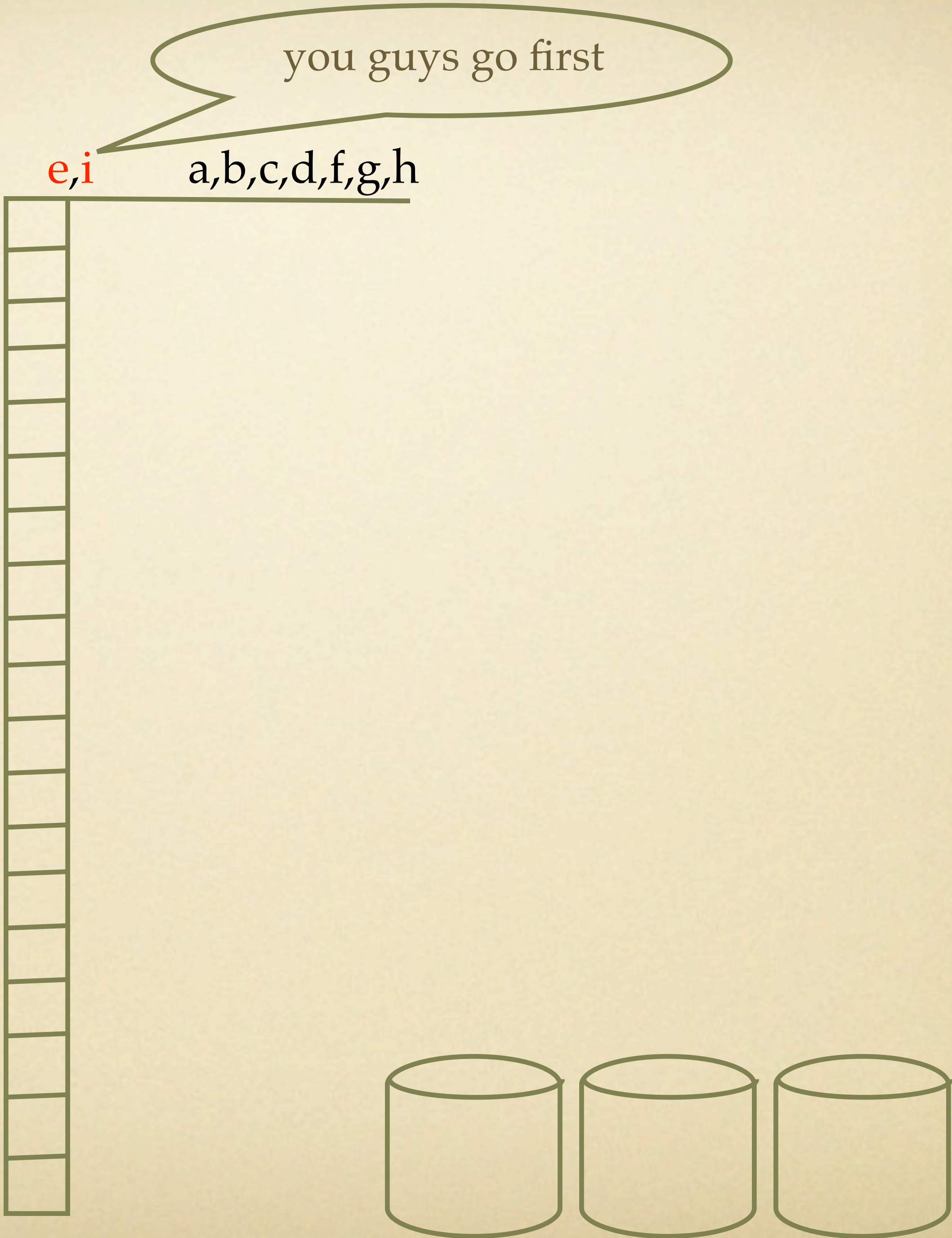
1. Each proc. picks a bin uniformly at random
2. Winners are candidates in lightest bin

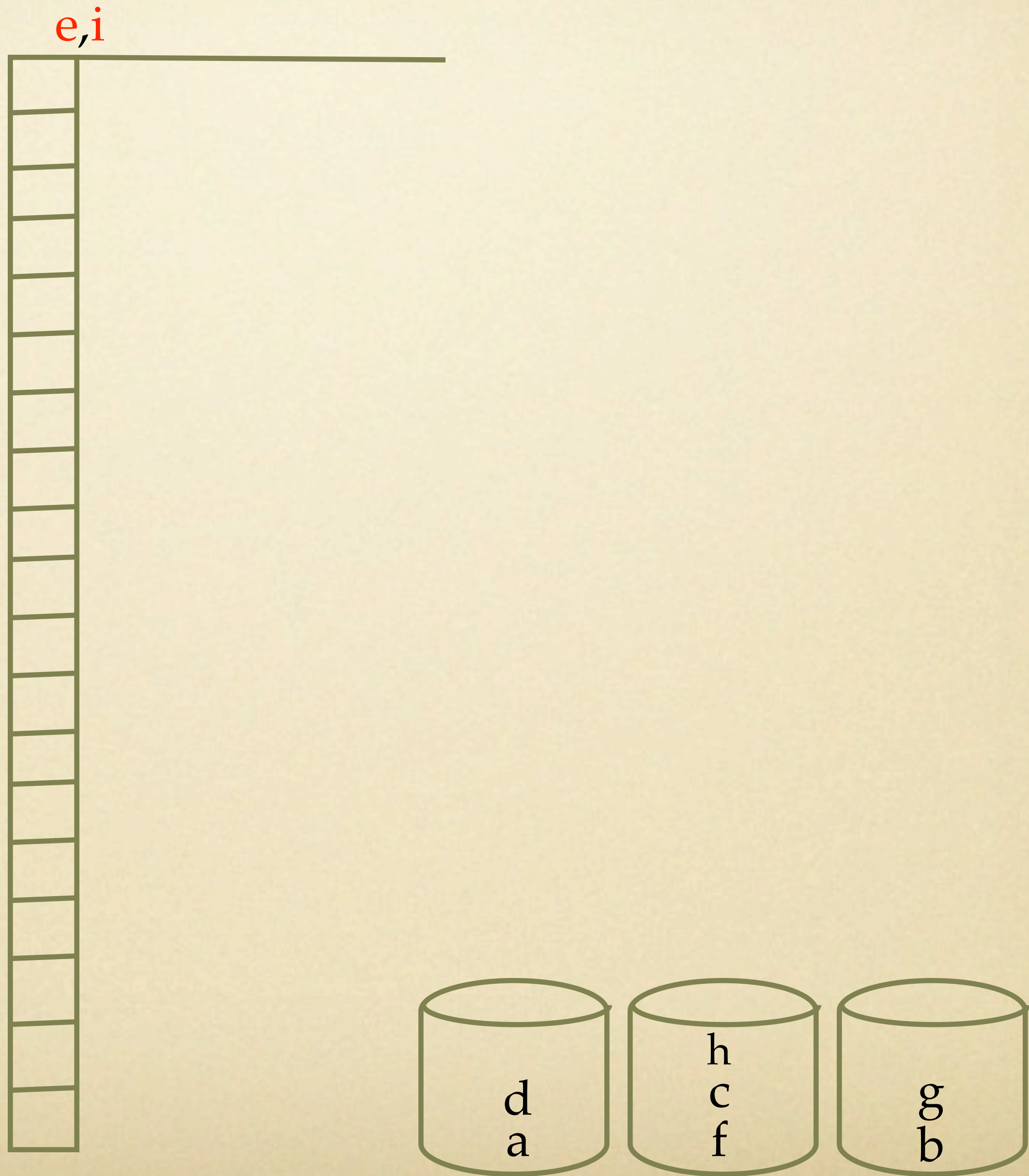
Feige

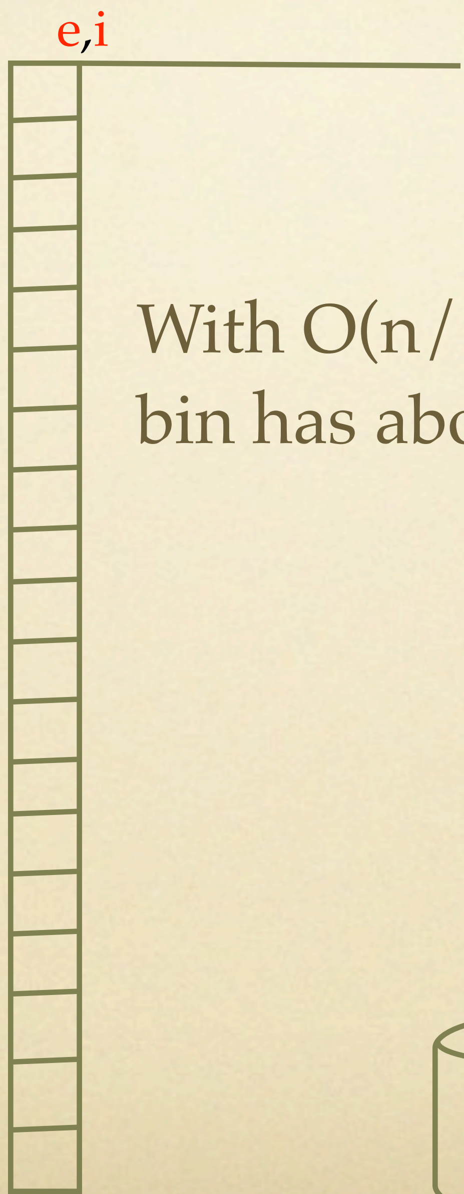
a,b,c,d,e,f,g,h,i



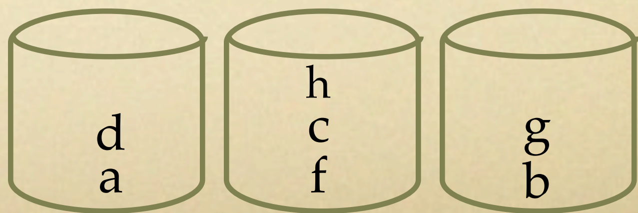

e,i
a,b,c,d,f,g,h

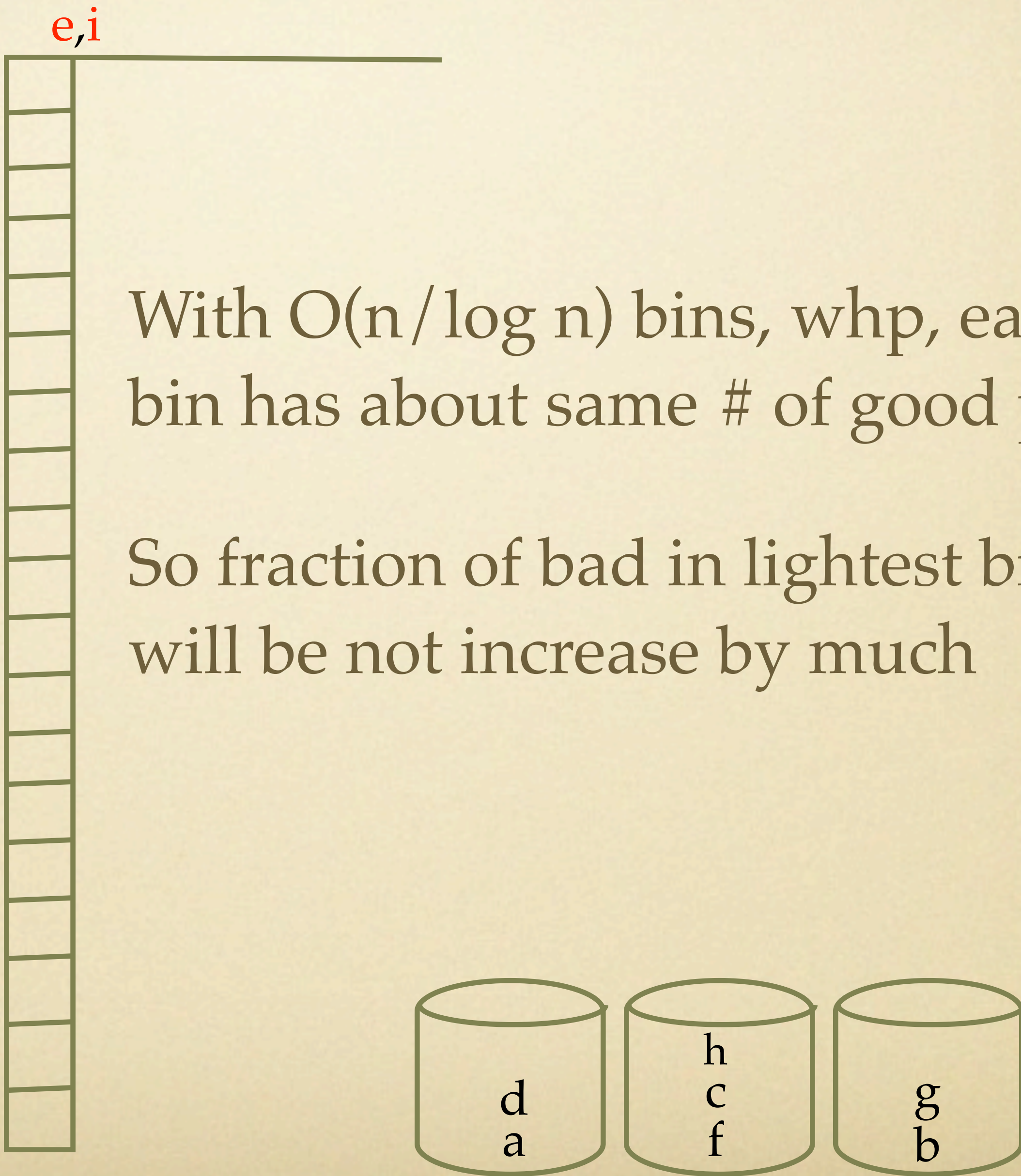

you guys go first
e,i
a,b,c,d,f,g,h


e,i
d
a
h
c
f
g
b



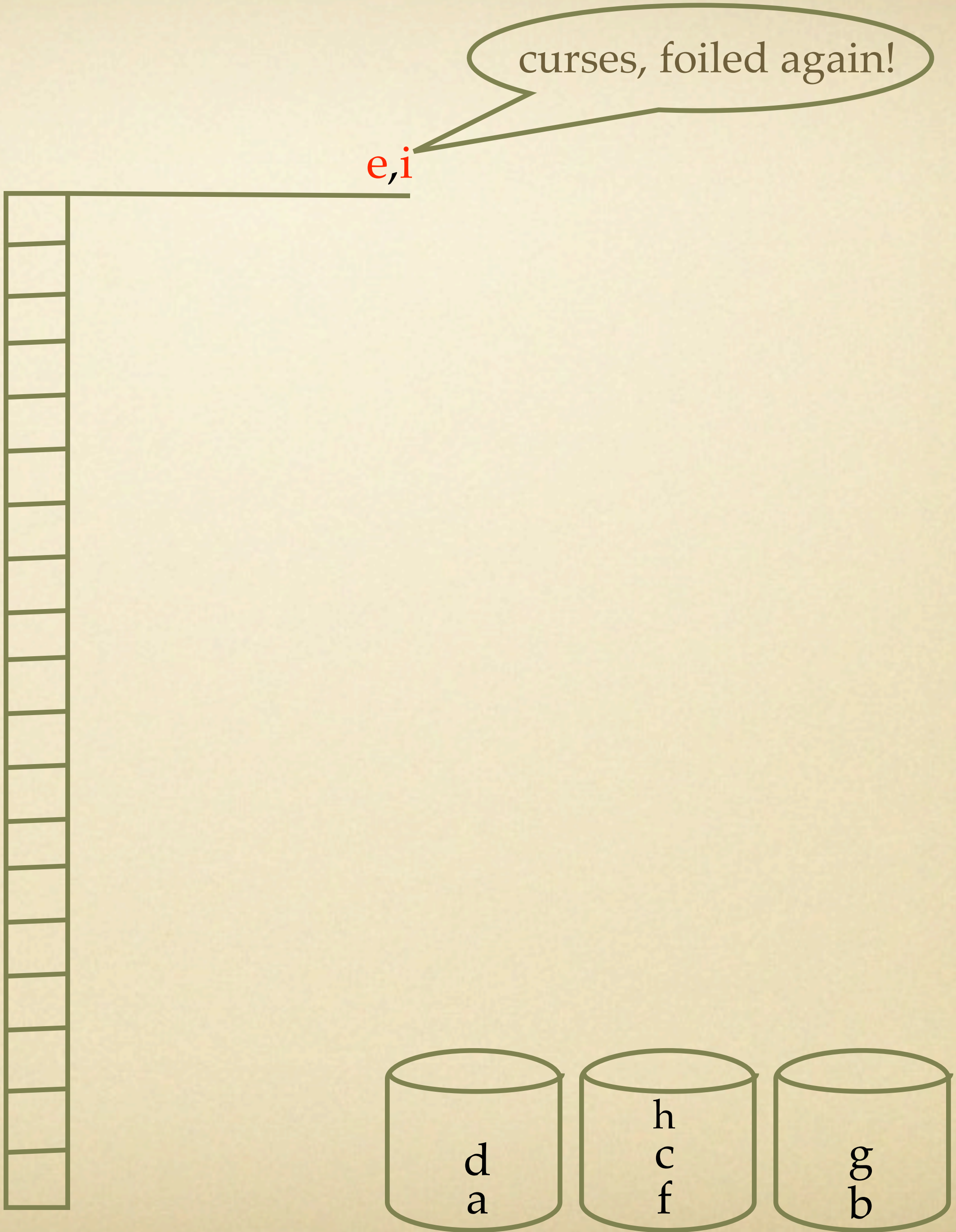
With $O(n/\log n)$ bins, whp, each bin has about same # of good procs

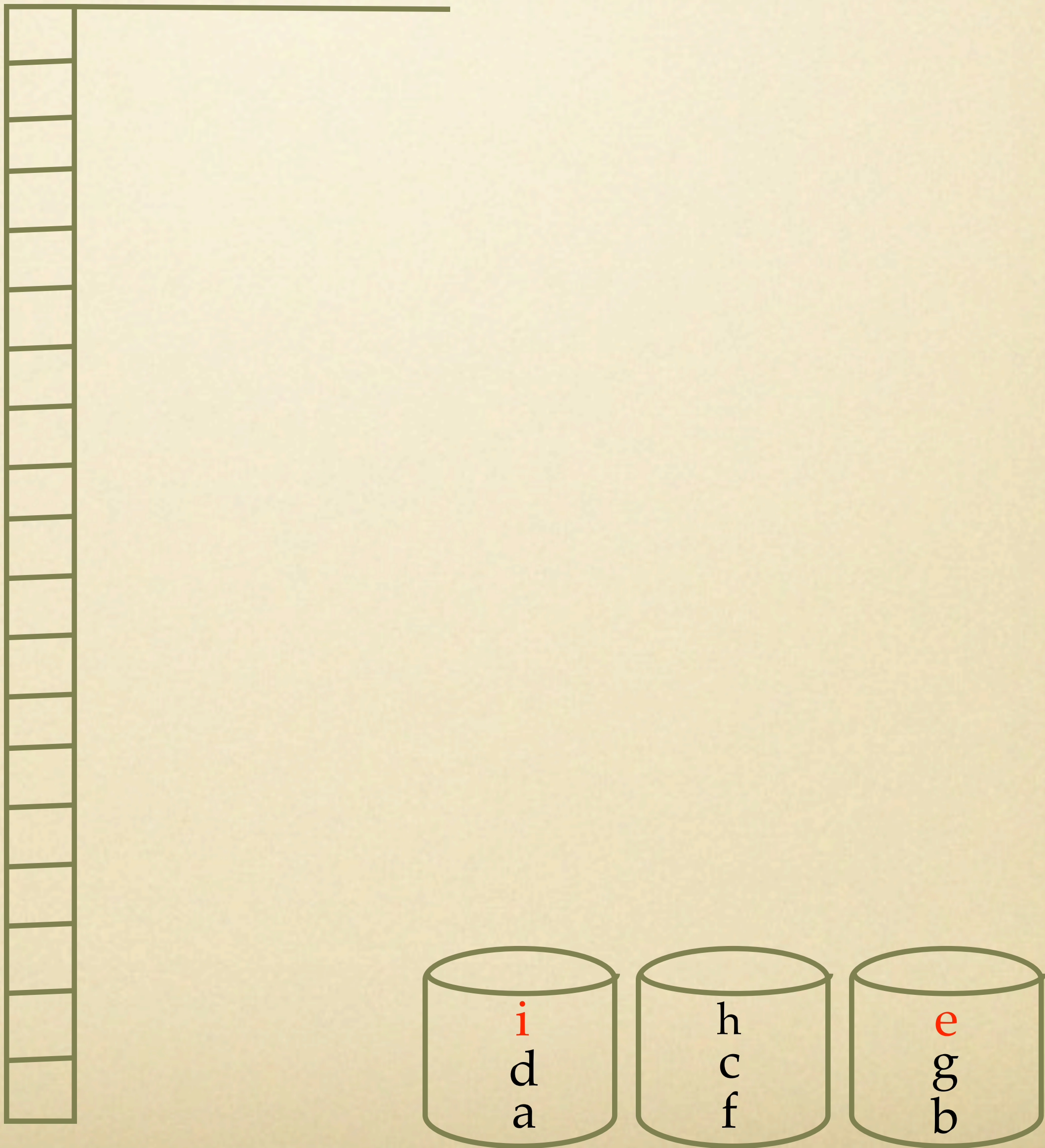


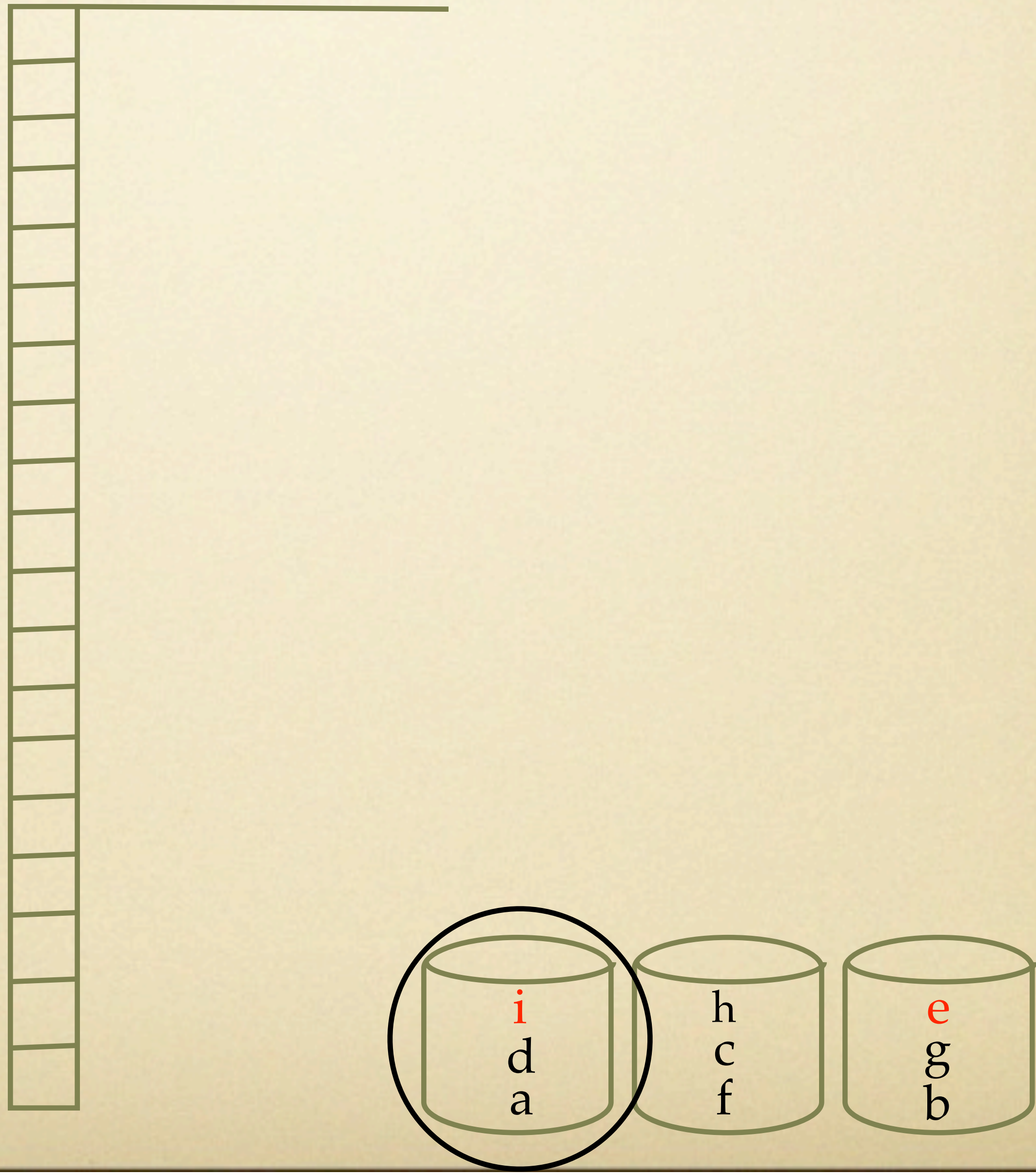


With O(n/log n) bins, whp, each bin has about same # of good procs

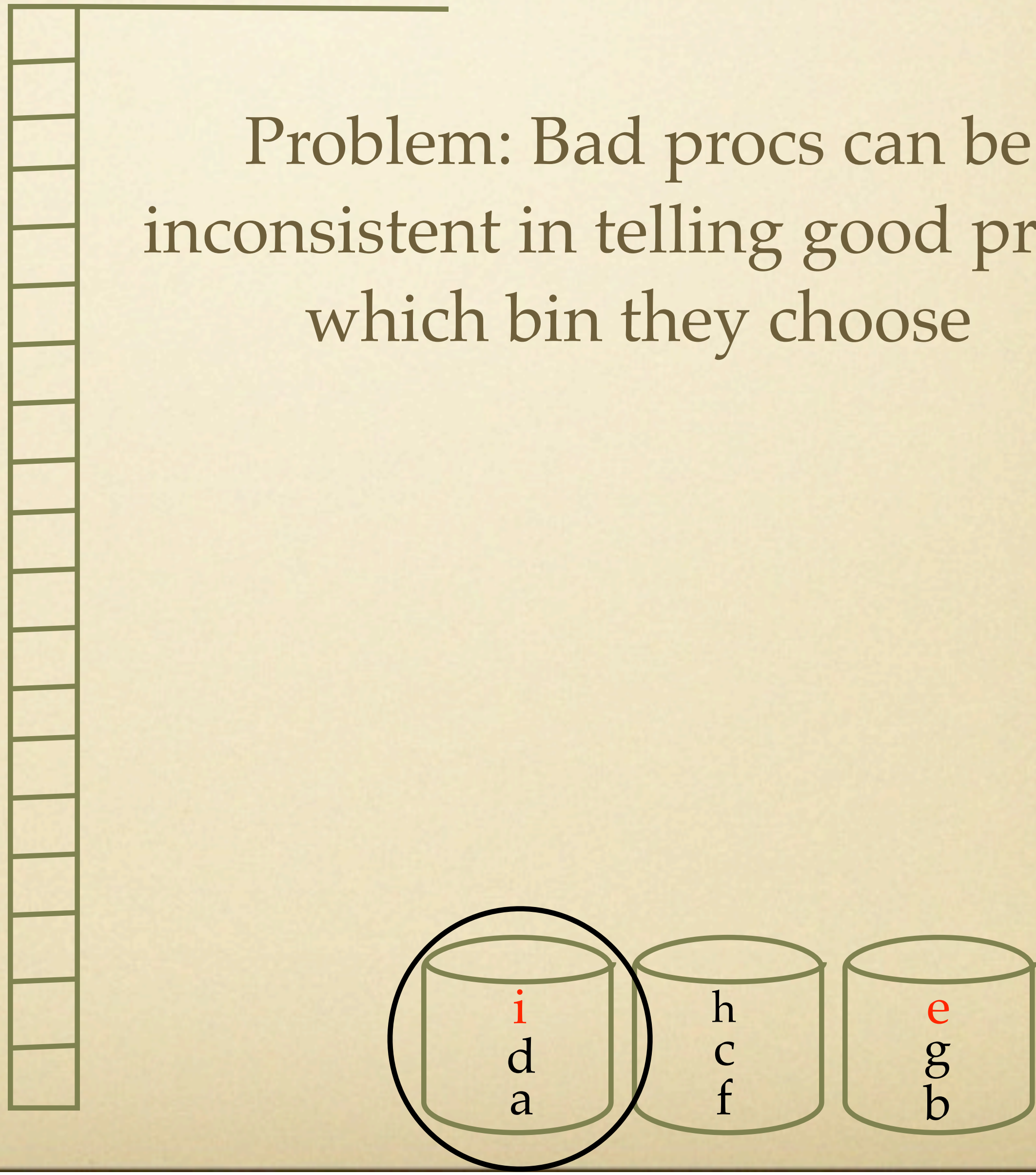
So fraction of bad in lightest bin will be not increase by much


curses, foiled again!
e,i
d
a
h
c
f
g
b


i
d
a
h
c
f
e
g
b

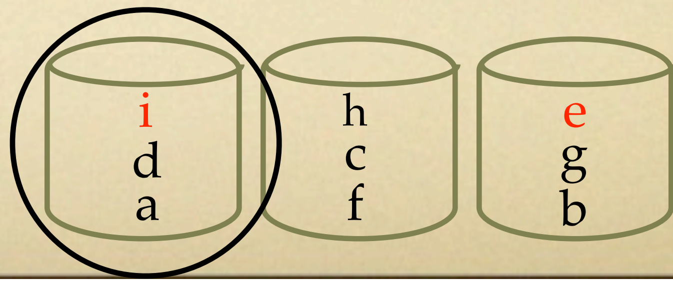

i
d
a
h
c
f
e
g
b

Problem: Bad procs can be inconsistent in telling good procs which bin they choose



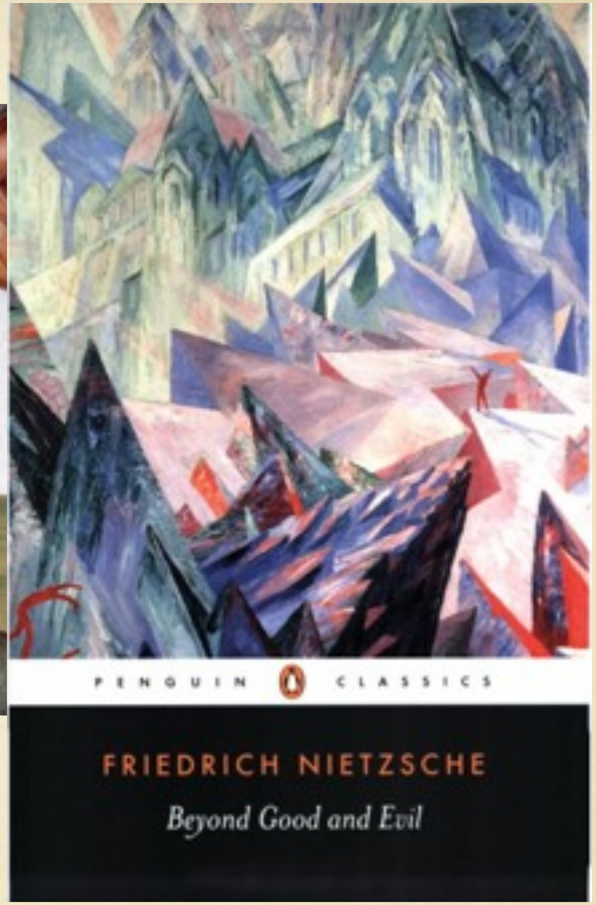
Problem: Bad procs can be inconsistent in telling good procs which bin they choose

Solution: Use Byznantine agreement to enforce a single bin choice for each proc!



A hard(er) fact:

Nobody follows instructions that aren't in their own best interest



A hard(er) fact:

Nobody follows instructions that aren't in their own best interest

Game Theory

- We assume all agent are selfish and rational
- **Nash equilibrium**: Situation where no player has any incentive to change its action
- Note: There may be more than one

Price of Anarchy(POA)

(Papadimitriou and Koutsoupias ’99)

- Social Welfare (SW) = Sum of costs of all players
- In most games, SW in Nash equil. is worse than SW with benevolent dictator
- POA measures that difference

POA (KP '99)

$$POA = \frac{\text{SW in Worst Equilibria}}{\text{SW with Benevolent Dictator}}$$

- Measures “tragedy of the commons” effect

POA

- POA can vary widely from one game to the other
- But there are many, many games with high POA

POA

- POA can vary widely from one game to the other
- But there are many, many games with high POA
- Problem: Can we reduce POA, without changing a game or injecting money or other resources?

Pollution Game

- Each player decides to pollute or not pollute
- Cost to a player is number of other players that pollute plus 2 if they do not to pollute

Pollution Game

- Each player decides to pollute or not pollute
- Cost to a player is number of other players that pollute plus 2 if they do not to pollute
- Nash Equilibrium: Everybody pollutes
- Benevolent Dictator (Optimal): Nobody pollutes

Pollution Game

- SW in Nash: n^2
- SW in Optimal: $2n$
- Price of Anarchy: $n/2$

Infinite Round

- Mediator: Advises each player not to pollute, until some player disregards advice. If this happens, from then on advise everyone to pollute.
- Result: Nobody pollutes!
- Significantly improves the SW

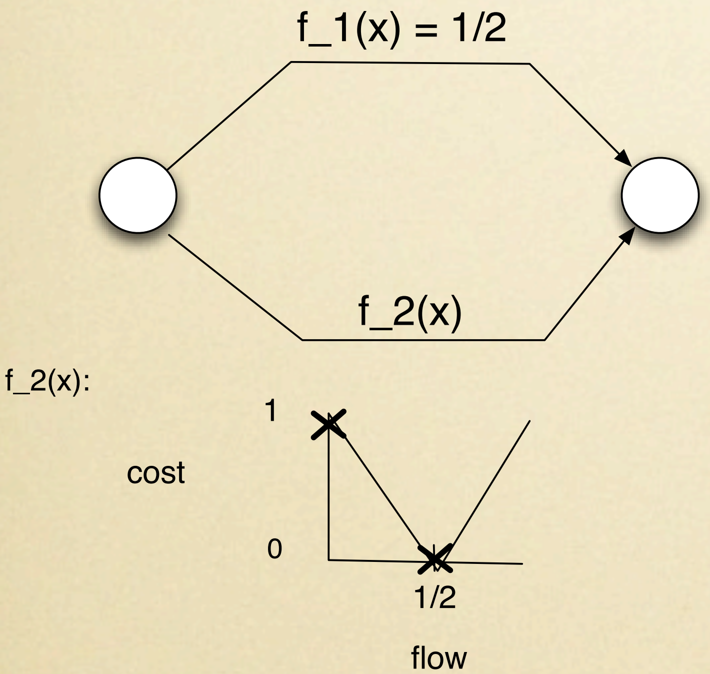
Mediator

- Mediator privately suggests an action to each player
- Players can ignore suggestions of mediator; they retain free-will and remain selfish
- Goal: Use mediator to improve SW

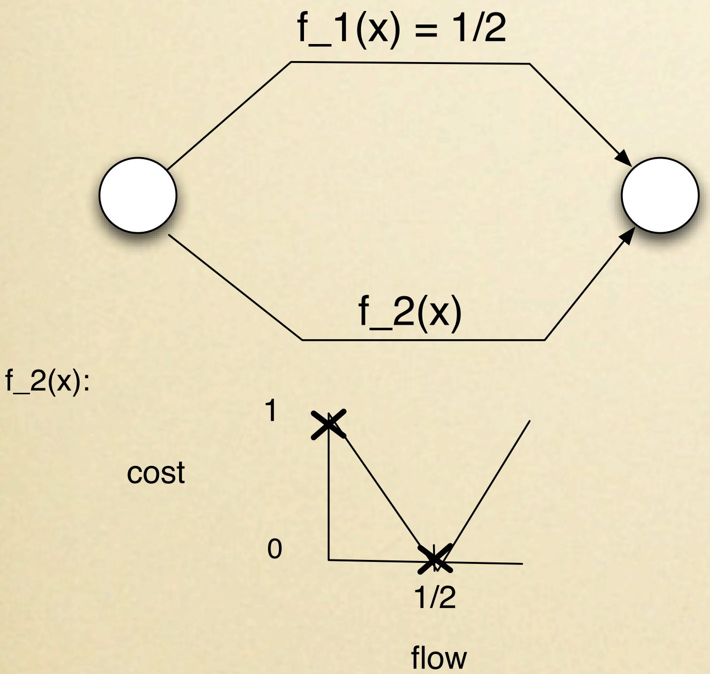
Mediator

- The mediator is an algorithm!
- The mediator might conceivably be a **randomized** algorithm
- A mediator may work even for a single round game!

El Farol Var.



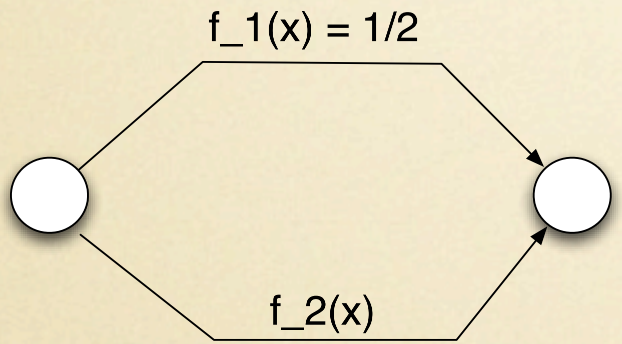
El Farol Var.



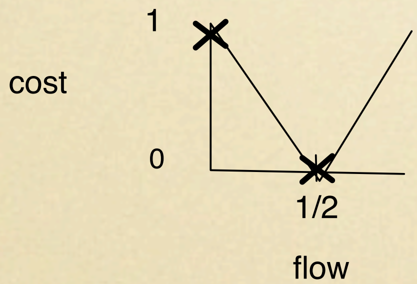
Mediator:

- With probability 1/3, tell all players to go up
- With probability 2/3, tell half the players to go up and half to go down

El Farol Var.



f_2(x):



Mediator:

- With probability 1/3, tell all players to go up
- With probability 2/3, tell half the players to go up and half to go down

Achieves S.W. of 1/3 vs 1/2 for the Nash

Mediator?

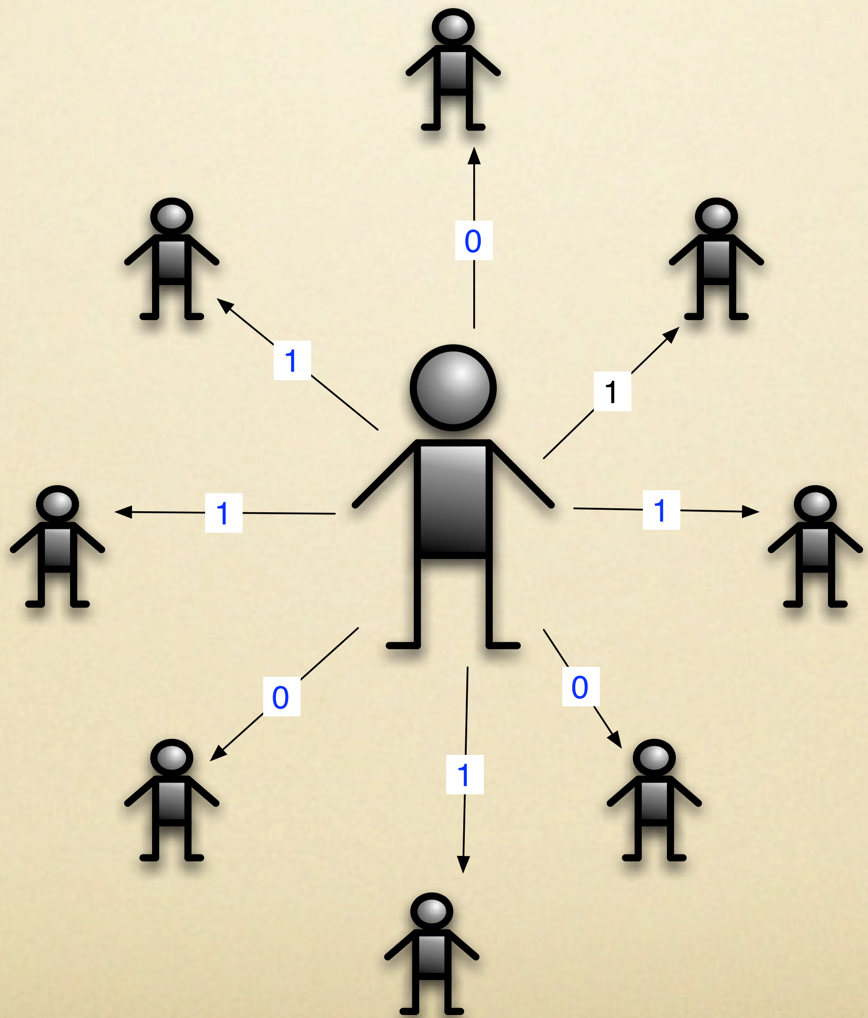
- Q: Where does the mediator come from?

Mediator?

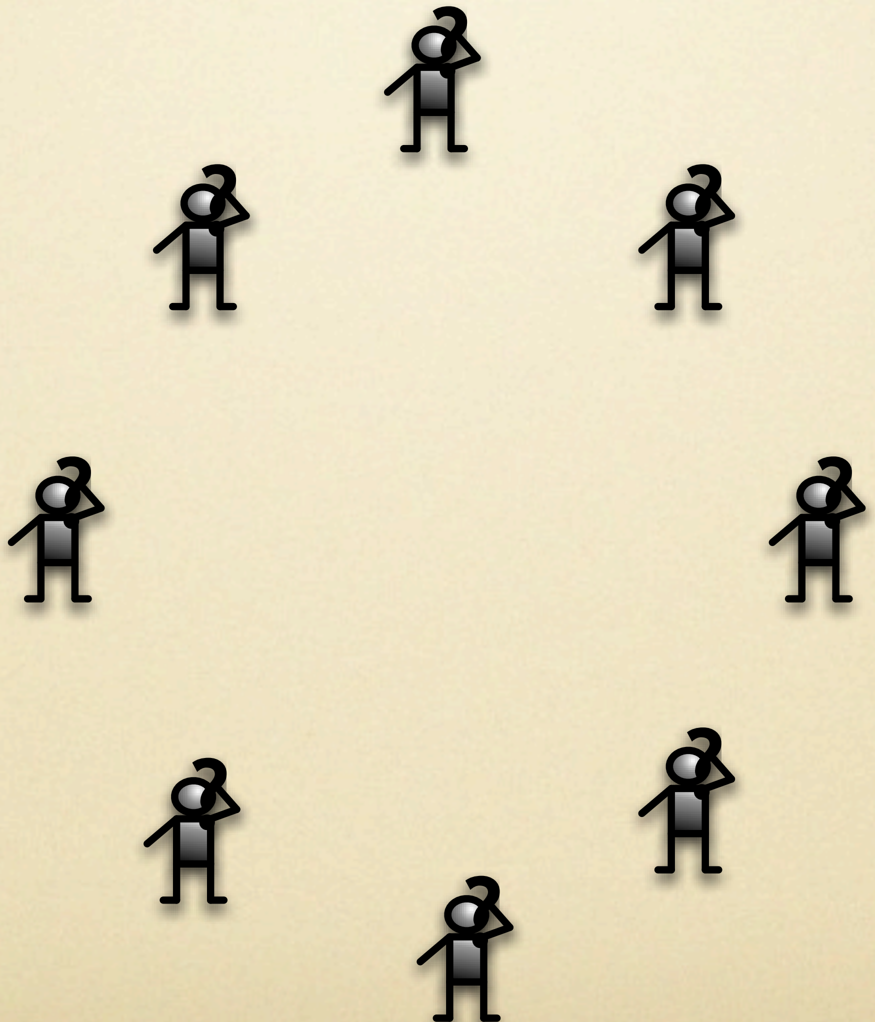
- Q: Where does the mediator come from?

“It is the final proof of God’s omnipotence that he need not exist in order to save us.” - Peter De Vries

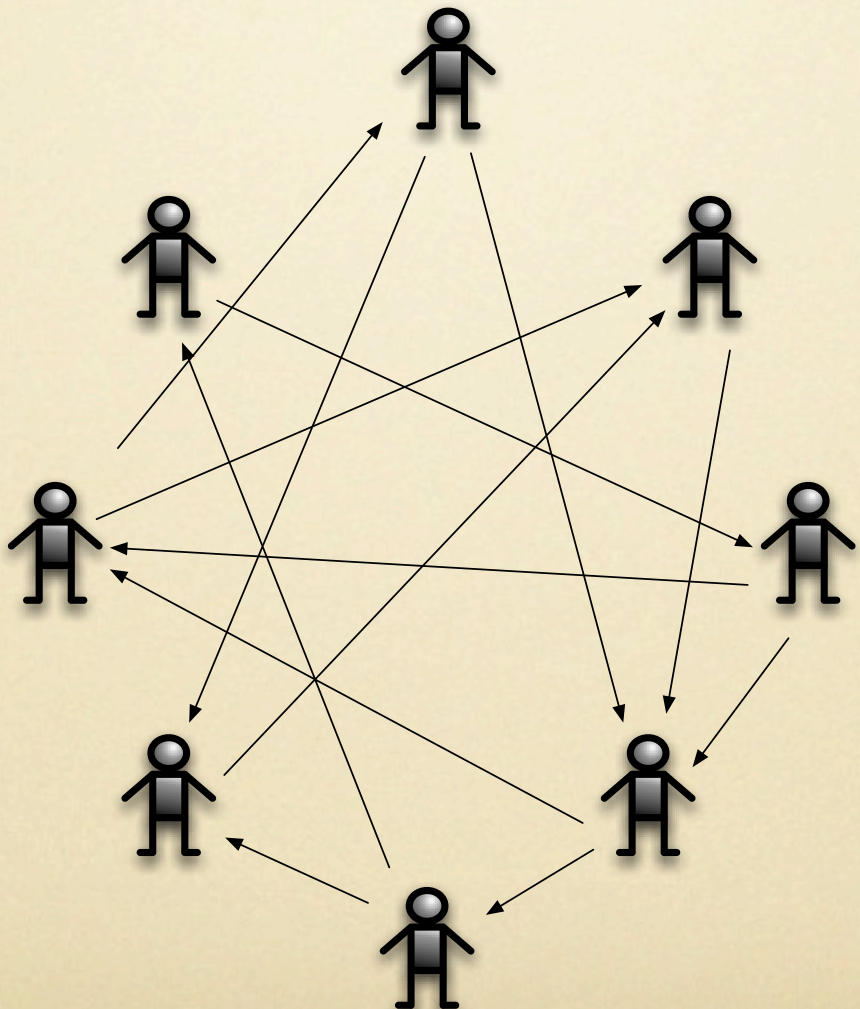
Mediator



No Mediator



No Mediator



Distributed Mediation

- A mediator can be implemented in a fully distributed manner by the players themselves ("cheap talk")
- Similar to cryptographic results on e.g. global coin toss and secure multiparty computation
- These algorithms make critical use of **Byzantine agreement**!

Auctions

- Similar techniques have been used to design completely distributed auctions
- No auctioneer!
- Nobody learns your bid unless you win!

Conclusion

We can still accomplish some goals even if not all agents blindly follow our instructions

Conclusion

We can still accomplish some goals even if not all agents blindly follow our instructions

We can accomplish some goals (just by offering advice) even when all agents have free-will

Conclusion

We can still accomplish some goals even if not all agents blindly follow our instructions

We can accomplish some goals (just by offering advice) even when all agents have free-will

However, the whole field is in its infancy and there still is a lot we don't understand about what is and what is not possible

Open Questions

- How efficiently can we perform Byzantine agreement?
- How efficiently can we implement a mediator?
- What properties must a game have in order for a mediator to be able to improve the social welfare?

Interested?

"When I talk about computer science as a possible basis for insights about God, of course I'm not thinking about God as a super-smart intellect surrounded by large clusters of ultrafast Linux workstations and great search engines. That's the user's point of view." - Donald Knuth

Interested?

Join Us!!!

Contact Info

- Questions, Ideas or thoughts?
- Google: Jared Saia to get my contact info
- I'm always interested in working with smart students