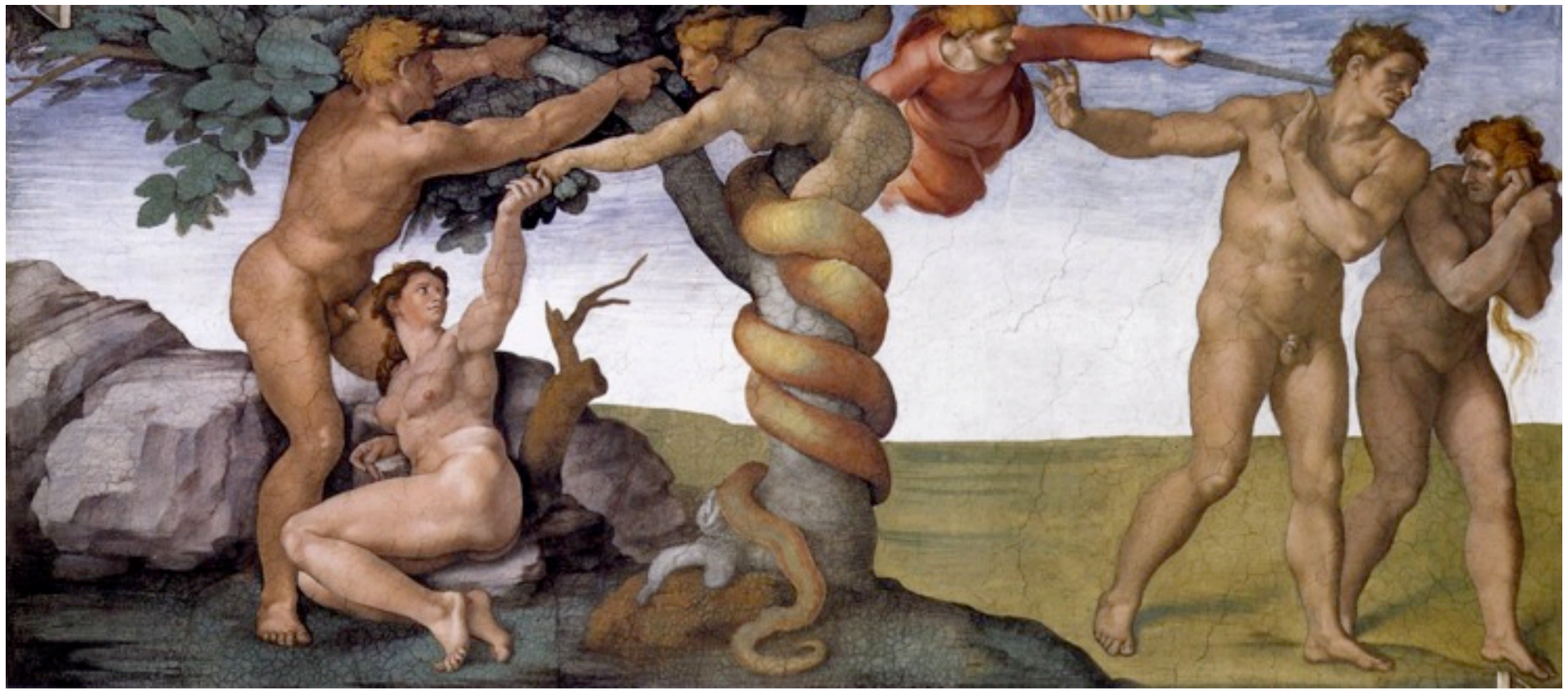
Scalable Mechanisms for Rational Secret Sharing

Jared Saia
with
Varsha Dani
Mahnush Movahedi
Yamel Rodriguez



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

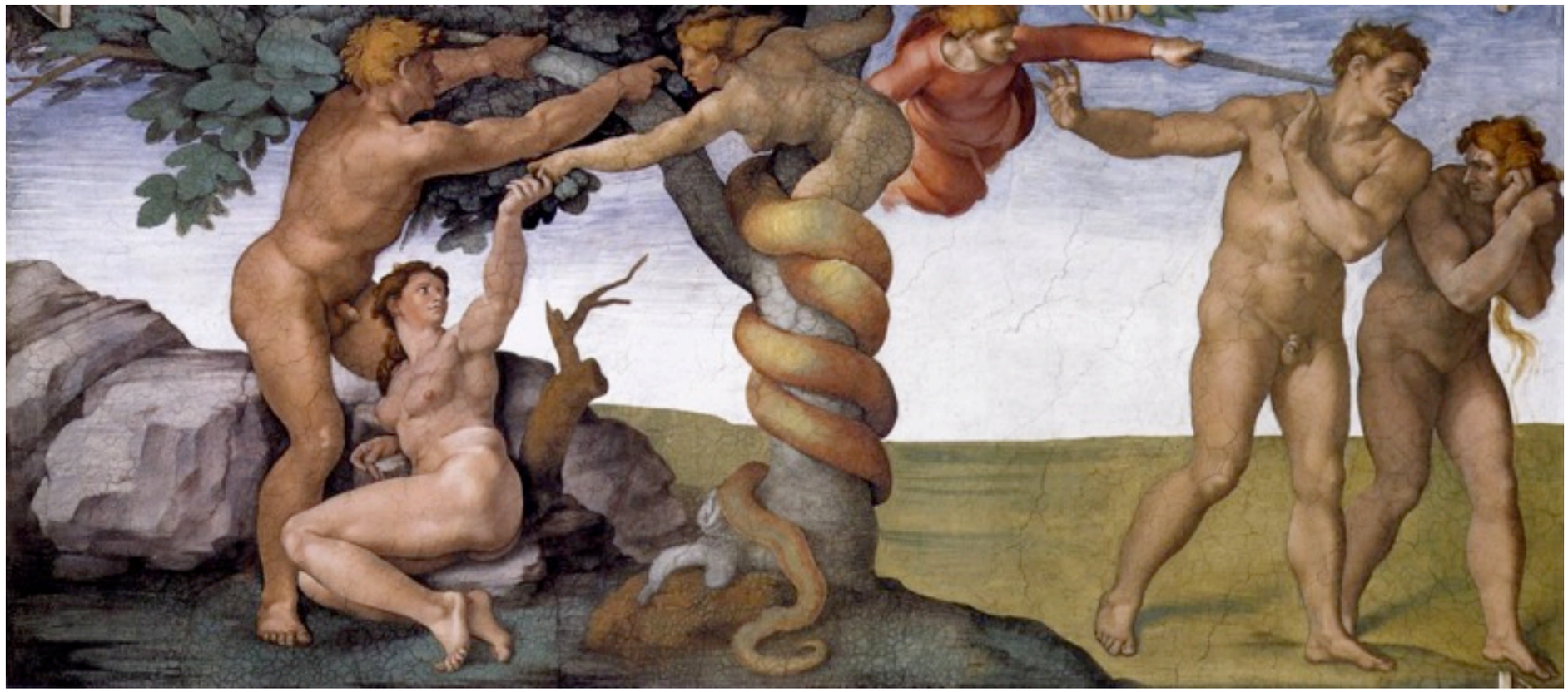
Good and Bad

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

Good: Follow instructions

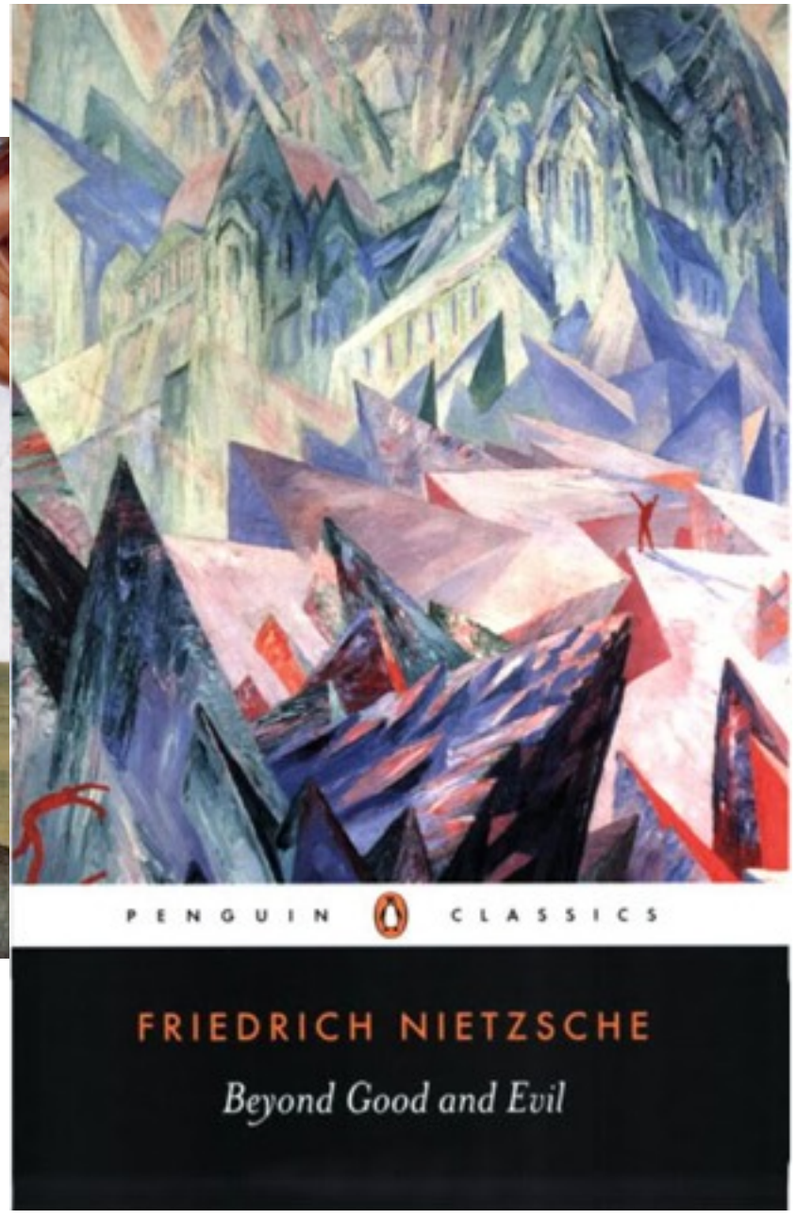
Bad: Don't follow instructions

Byzantine agreement, Secure Multiparty Computation, etc.



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

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: $n/2$

Mediator

- Mediator privately suggests an action to each player

- Players may ignore suggestions of mediator; they retain free-will and remain selfish

- Goal: Use mediator to improve SW

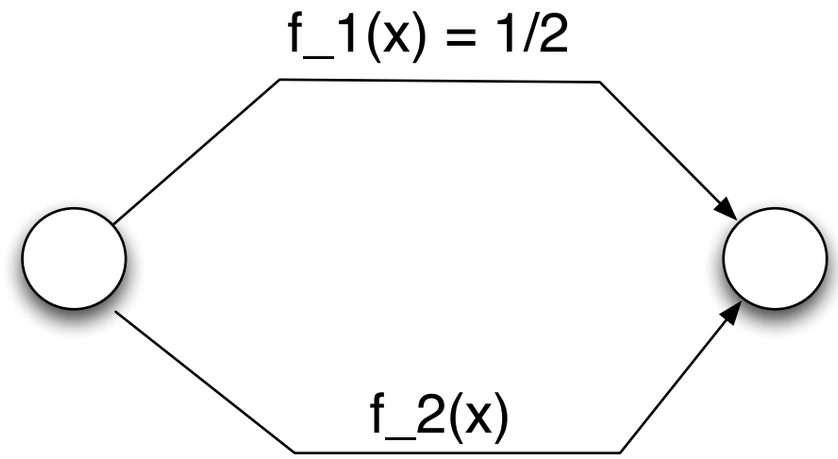
Pollution Game

- 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

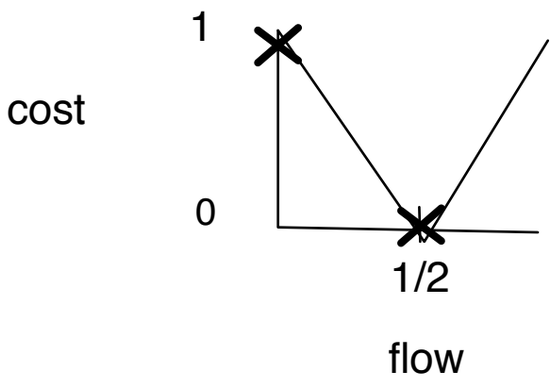
Mediator

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

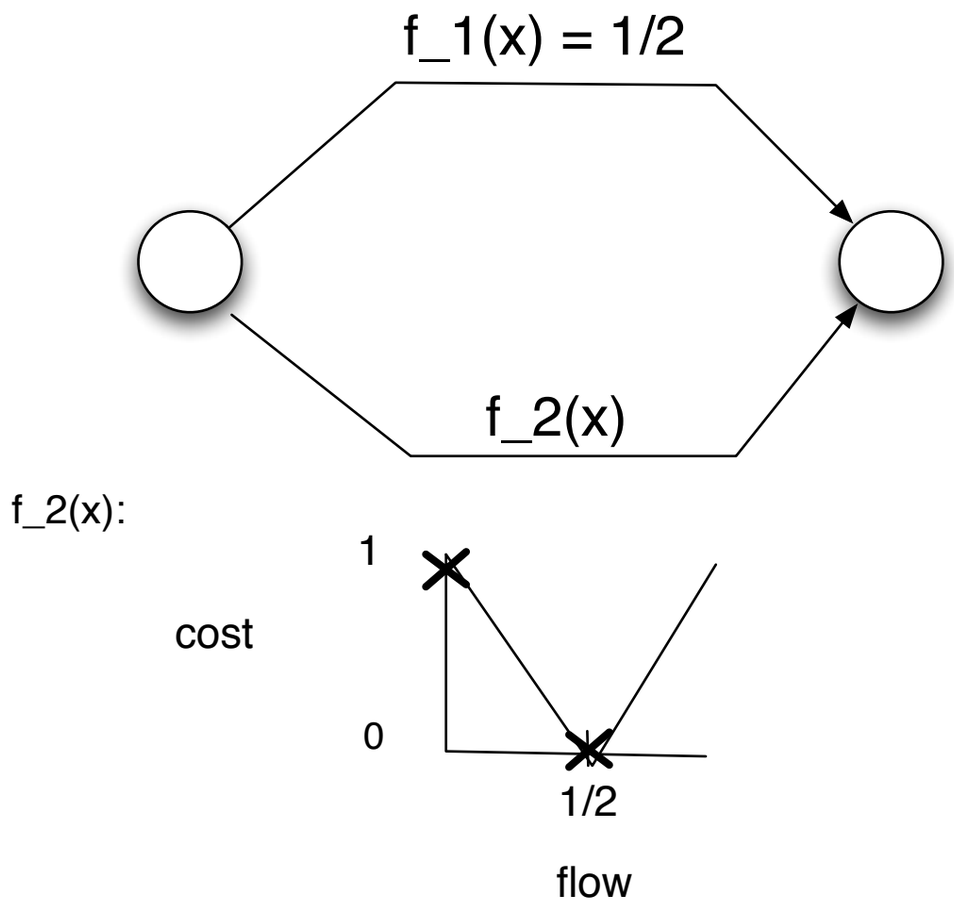
El Farol



f_2(x):



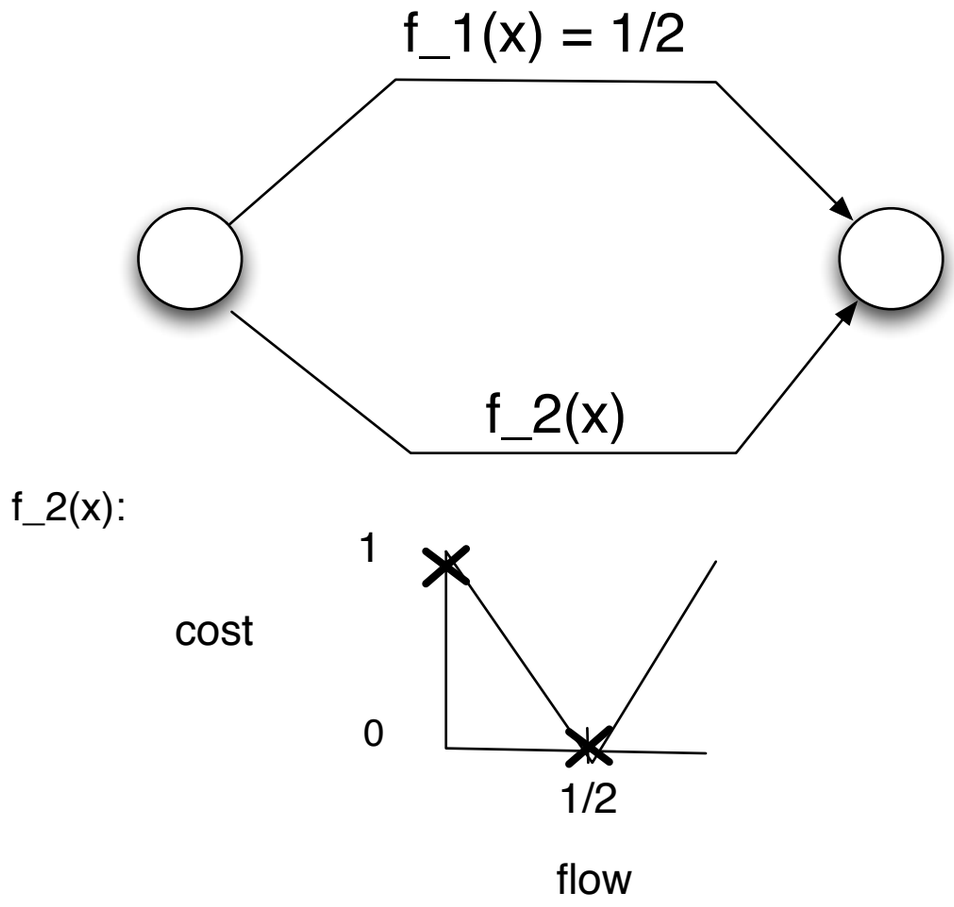
El Farol



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



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

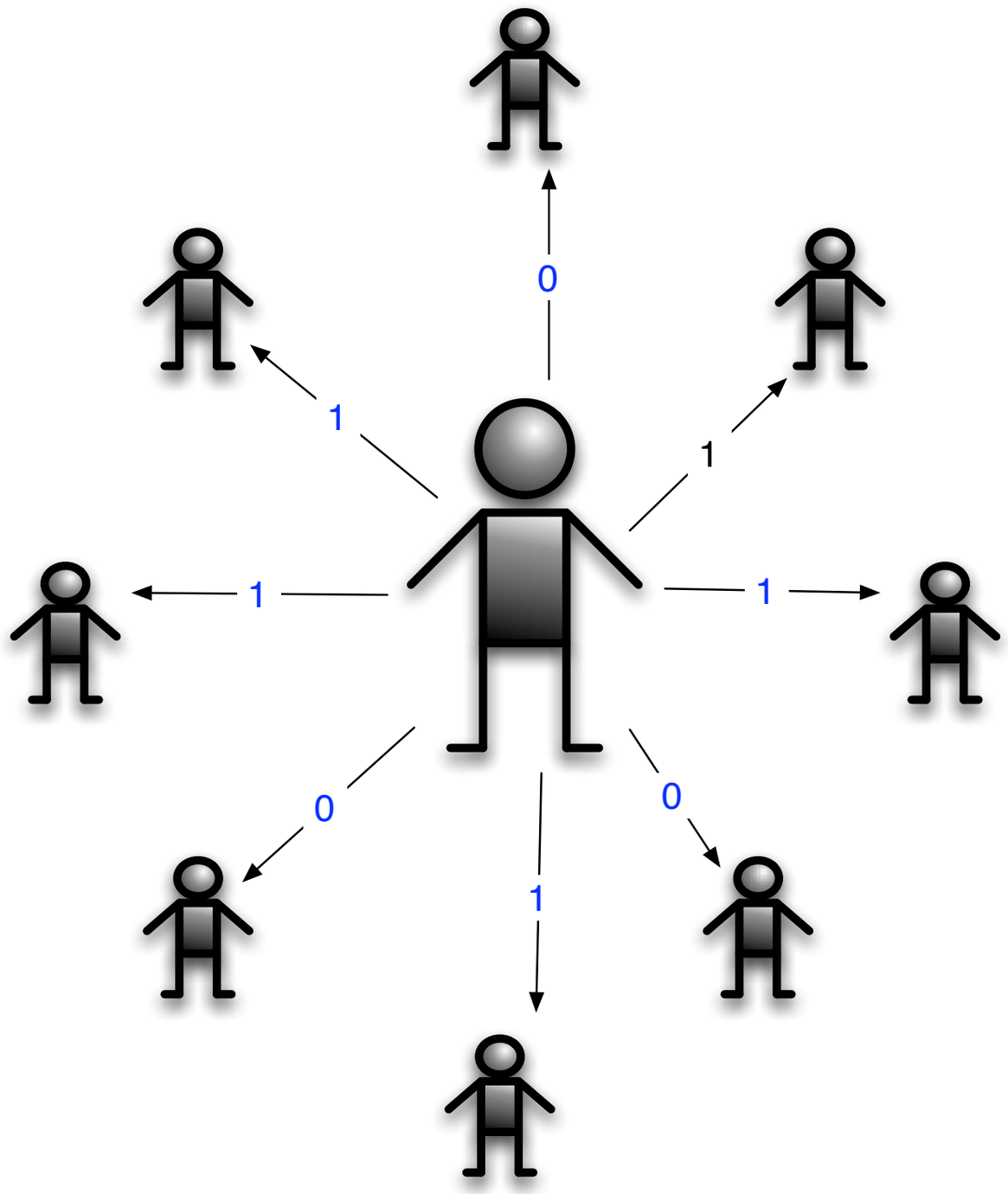
Where does the mediator come from?

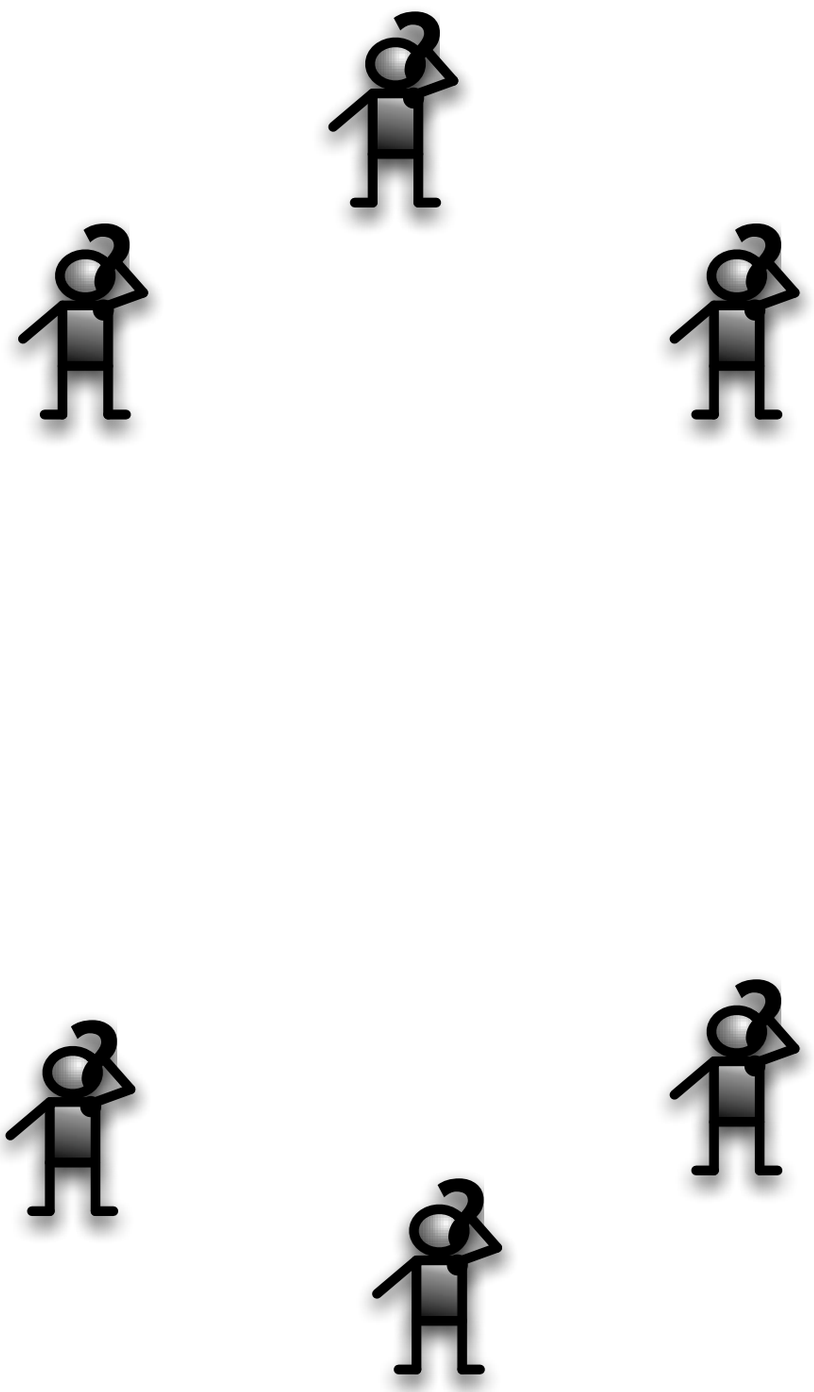
Mediator

Where does the mediator come from?

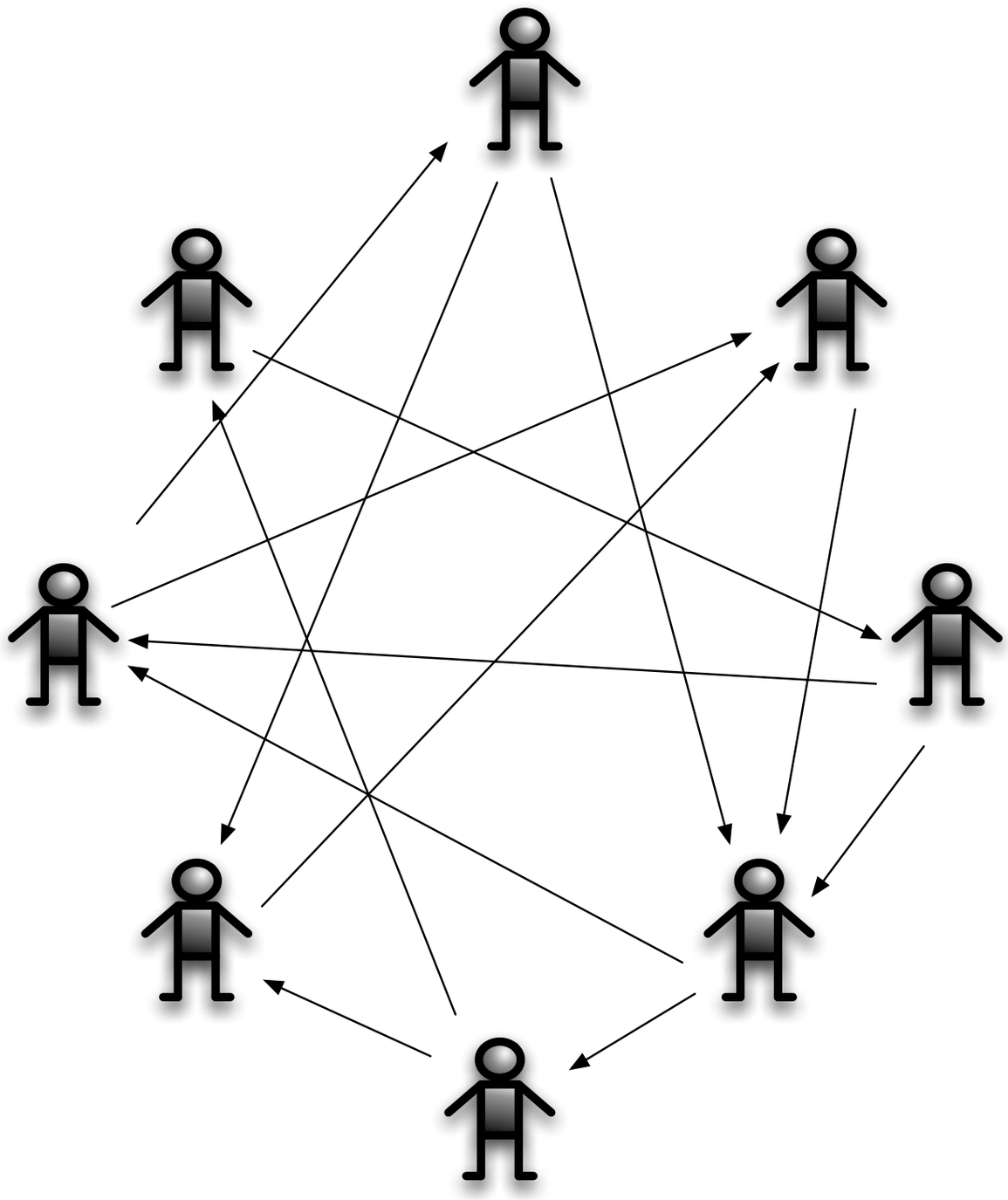
“It is the final proof of God's omnipotence that he need not exist in order to save us.” -
Peter De Vries

Mediator





Mediator



Mediator

Goal: Implement a mediator in a fully decentralized manner ("cheap talk")

Previous work: Abraham et al. ['06,'08], Lysyanskaya & Triandopoulos ['06]

Mediator

Goal: Implement a mediator in a fully decentralized manner ("cheap talk")

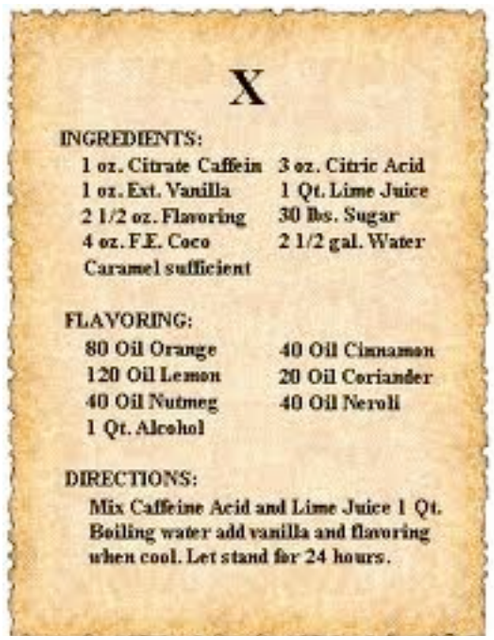
Previous work: Abraham et al. ['06,'08], Lysyanskaya & Triandopoulos ['06]

Problem: not *scalable* i.e. require all-to-all communication

Goal: *Scalable* protocol for secret sharing

Secret Sharing?

- *m-out-of-n secret sharing*: m ≤ n : A secret is distributed among n players such that m together can reconstruct the secret, but fewer than m obtain no information
- Shamir, Blakley [1979]



Shamir's Scheme

- Embed the secret S in a finite field
- Choose a random polynomial P of degree m-1 with P(0) = S
- Shares are P(1), P(2), ..., P(n)
- Any m of the points (i, P(i)) can be used to interpolate the polynomial P

Adding Game Theory

- Shamir's scheme assumes players follow instructions.
- What if instead they are selfish? Each wants to learn the secret, but doesn't want the others to learn it.
- Goal: Nash Equilibrium protocol that ensures when (and only when) there is a quorum, everyone learns the secret

Formal problem

- n rational players
- utility functions described by 3 parameters
  u_+ : utility for learning alone
  u : utility for learning (others learn too)
  u_- : utility for not learning
- GOAL: NE protocol that ensures that all players learn the secret.

Prior Work

- Introduced by Halpern & Teague ['04]
- Abraham, Dolev, Gonen & Halpern ['06]; Gordon & Katz ['06]. Cryptographic.
- Kol & M. Naor ['08] -- Non-cryptographic.

Modes of Communication

- Most previous work assumes simultaneous broadcast channels
- KN also allow non-simultaneous broadcast
- Broadcast is bad model for large network. Instead want point-to-point (private) channels.

Modes of Communication

- Most previous work assumes simultaneous broadcast channels
- KN also allow non-simultaneous broadcast
- Broadcast is bad model for large network. Instead want point-to-point (private) channels.

Modes of Communication

- Most previous work assumes simultaneous broadcast channels
- KN also allow non-simultaneous broadcast
- Broadcast is bad model for large network. Instead want point-to-point (private) channels.

Idea: Fake Rounds

- Problem: If everyone sends me their shares, I learn the secret without revealing my share (others don't learn, not NE)
- Idea: Have "fake" rounds to catch cheaters. If anyone cheats, everyone quits, nobody wins. Real secret is reconstructed on a random round.

Idea: Different Lengths

Problem: If know # rounds, no penalty for defaulting on the last round. So second to last round is effectively the last, so no penalty for defaulting on second to last round, etc. ("Backward Induction")

Idea: Different Lengths

Problem: If know # rounds, no penalty for defaulting on the last round. So second to last round is effectively the last, so no penalty for defaulting on second to last round, etc. ("Backward Induction")

Idea [KN]: Use shares of different lengths for players. Can't tell from your input whether you have a shorter or longer share. No base case for backward induction!

Scalable

- n rational players who can communicate over private channels
- utility functions described by 3 parameters
  u_+ : utility for learning alone
  u : utility for learning (others learn too)
  u_- : utility for not learning
- GOAL: NE protocol that ensures that all players learn the secret. Should be scalable, i.e. polylog(n) messages.

Alas...

- There is none. [KN '08] (specifically, no NE for non-simultaneous broadcast)
- Relax the requirements: still want private channels & scalability, but settle for weaker equilibrium concept.

ε-Nash Equilibrium

- A set of strategies such that if all other players follow, no player can gain more than ε in expectation by deviating.
- We will actually get something stronger: no player can gain more than ε in expectation in any round, even conditioned on history (ε-everlasting equilibrium)

The Mechanism

- Two components: 1) dealer's protocol to create inputs; and 2) players' protocol once the game starts
- Initially: n-out-of-n; Later: m-out-of-n
- Initially: success with probability $1-\delta$; Later: success w.h.p

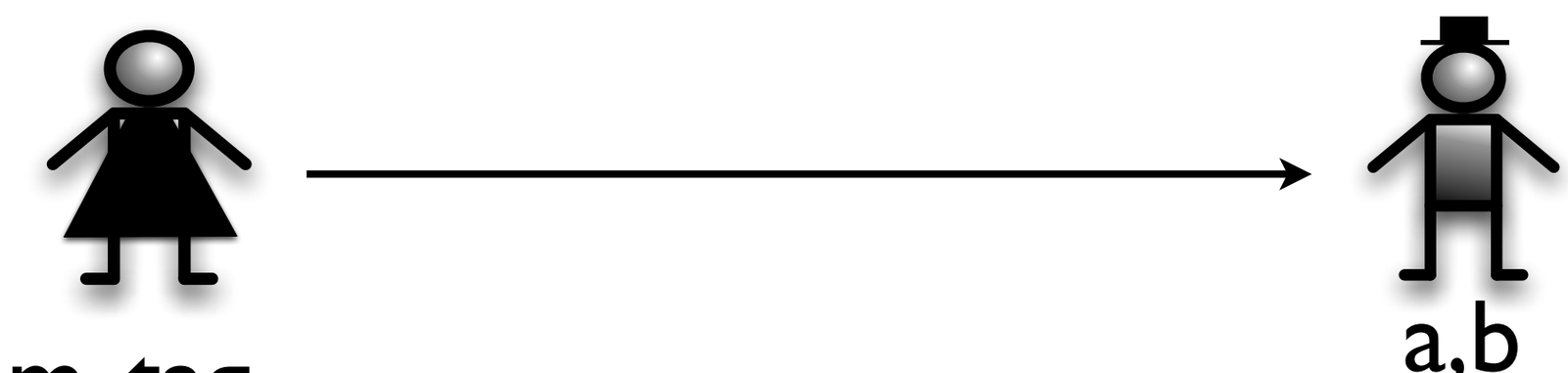
Tag and Hash

Dealer wants Alice to send a message m to Bob at a point in the future

Wants Bob to be able to verify the message is correct when received

Doesn't want Bob to learn anything about the message *before* it's received

Tag and Hash



a & tag are random non-zero elements in F_q

Alice sends m and tag to Bob

Bob checks that b = a*m + tag

Prob(success in faking a message) ≤ 1/(q-1)

Dealer's Protocol

- Embed secret in large finite field F_q
- Choose a player at random: "short player"
- X, Y, Z i.i.d. ~ Geom(β)
  X : position of secret
  Y : padding on short input
  Z : additional padding on long input
- Long players get lists of length X+Y+Z.
  Short player gets list of length X+Y.

Lists

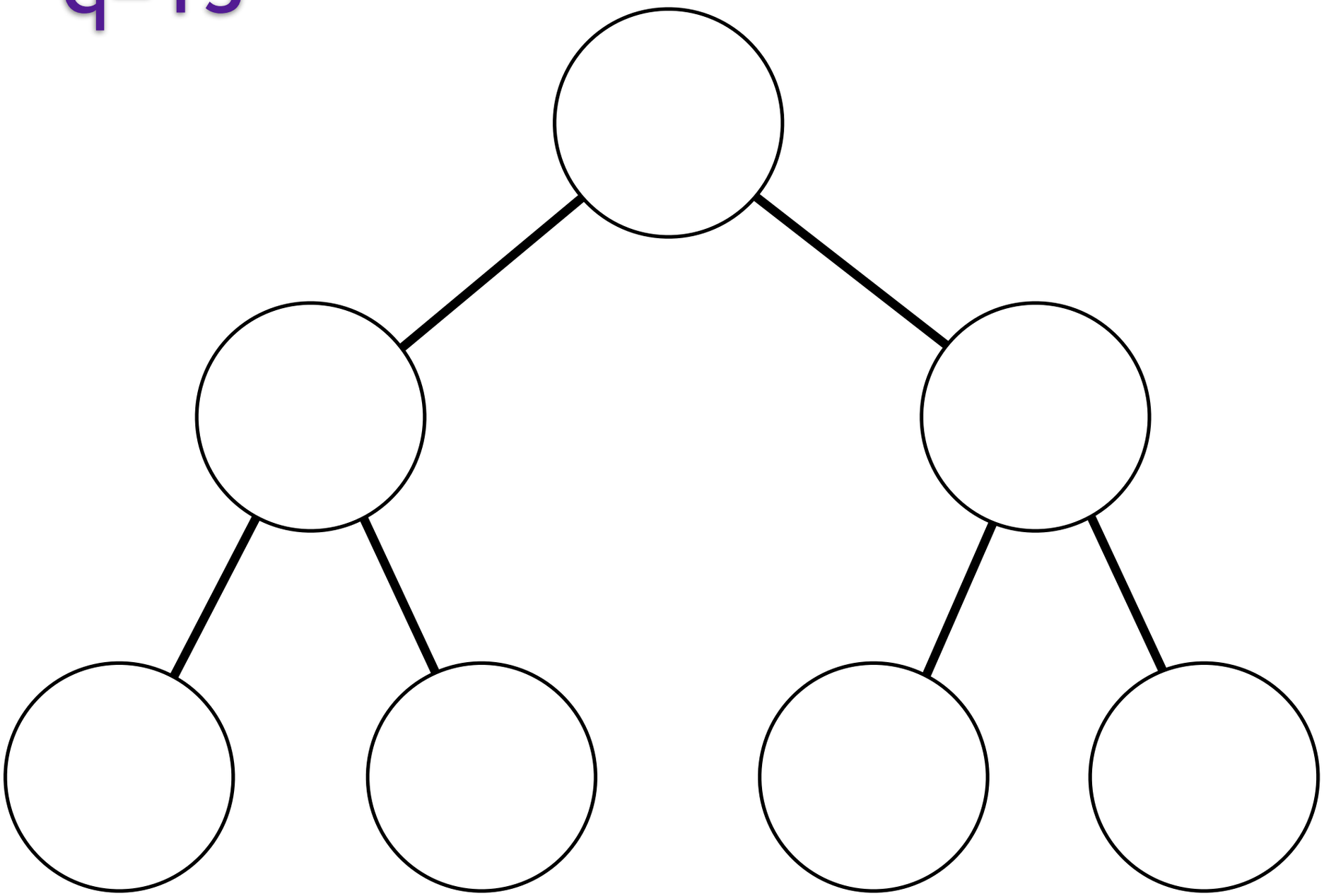
- A player's input:
  - List of potential (masked) secrets
  - List of tags (to authenticate messages you send)
  - List of hashes (to verify messages you receive)
  - List of recursive shares for
    - masks (for next round)
    - indicators (0 if this is round X)

Recursive Shares

- Create a complete binary tree with n leaves.
- Assign a player to each leaf.
- Assign all but one player to internal nodes.
- Perform recursive (Shamir-like) secret sharing starting at the root node

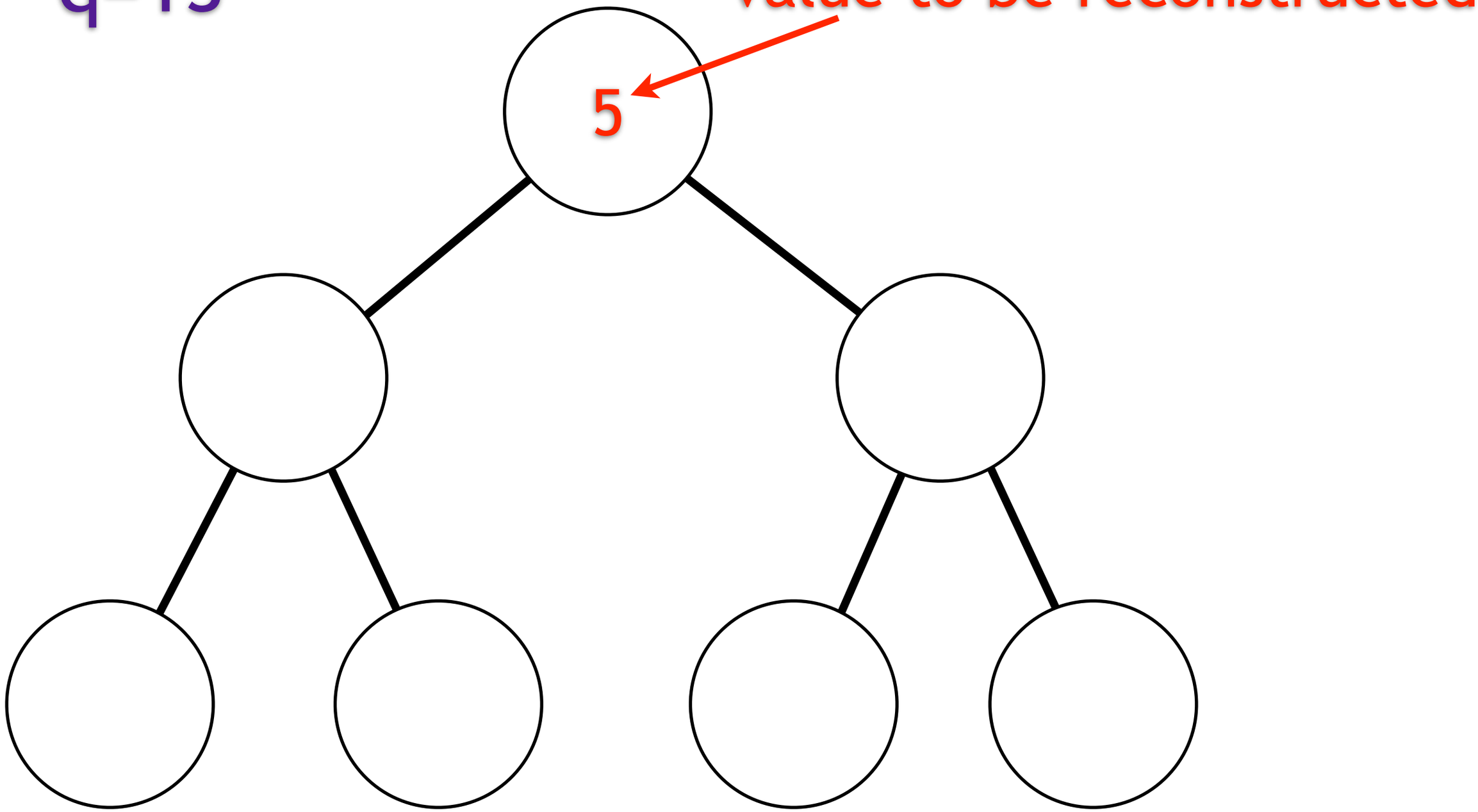
Making the Shares

q=13



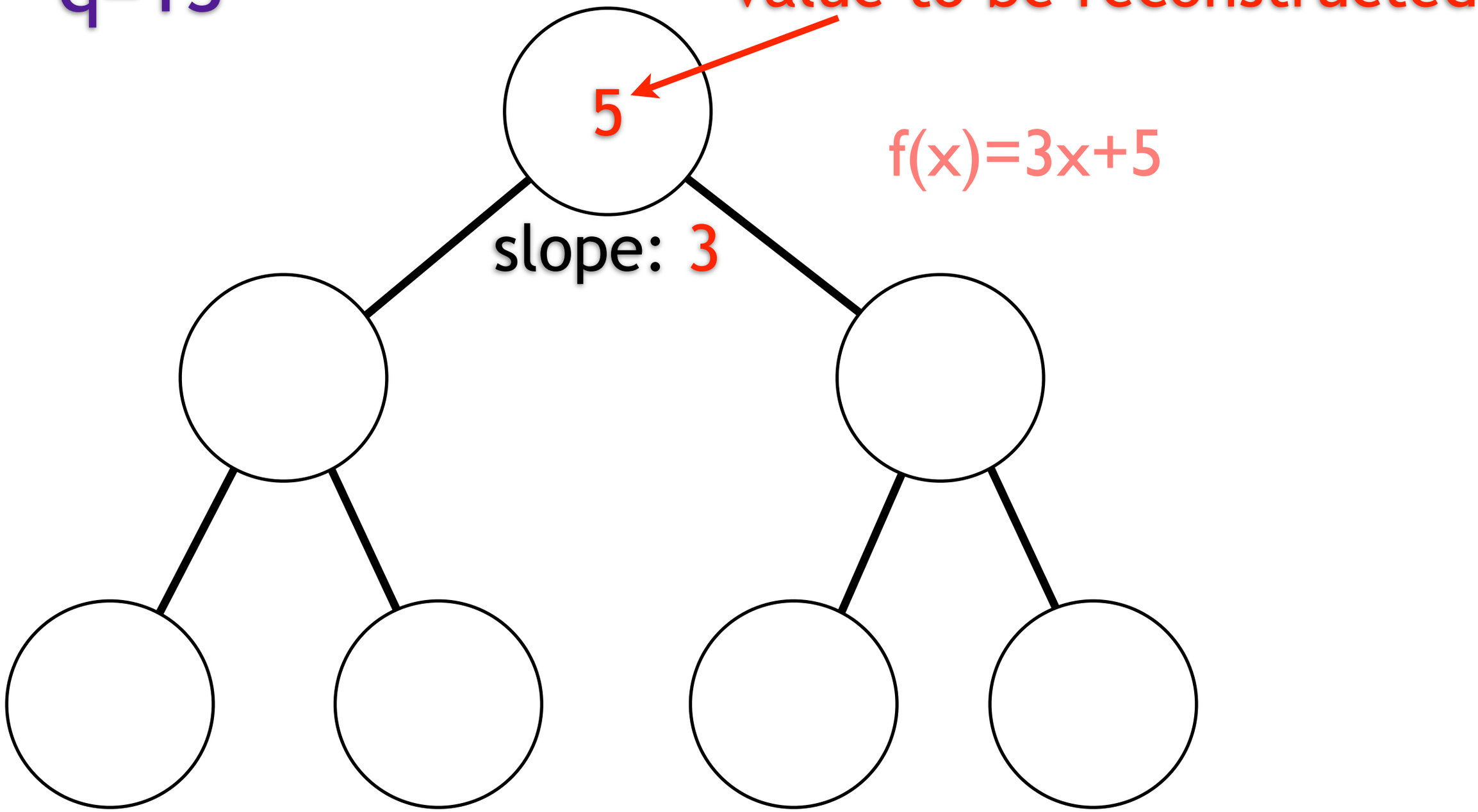
Making the Shares

q=13



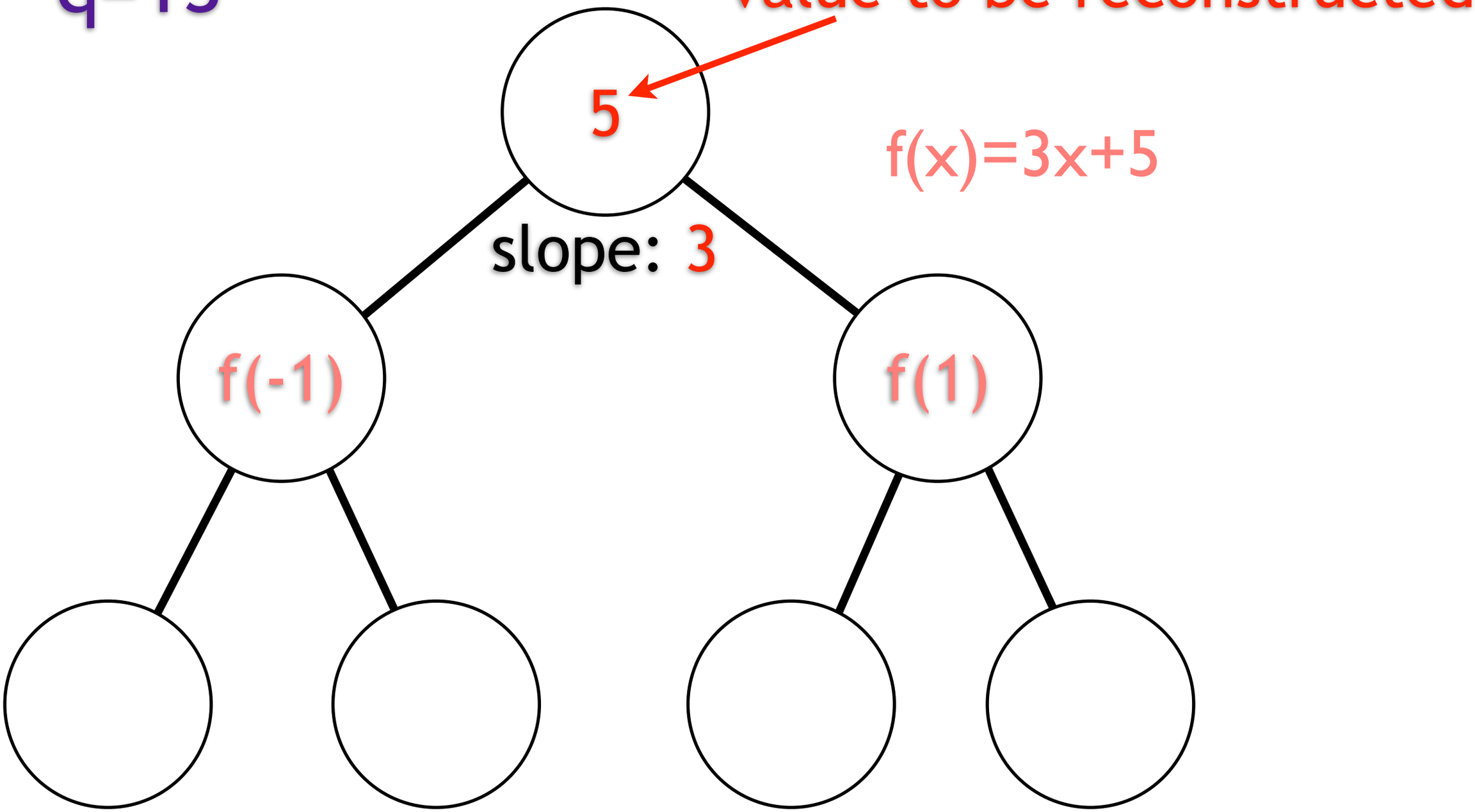
Making the Shares

q=13

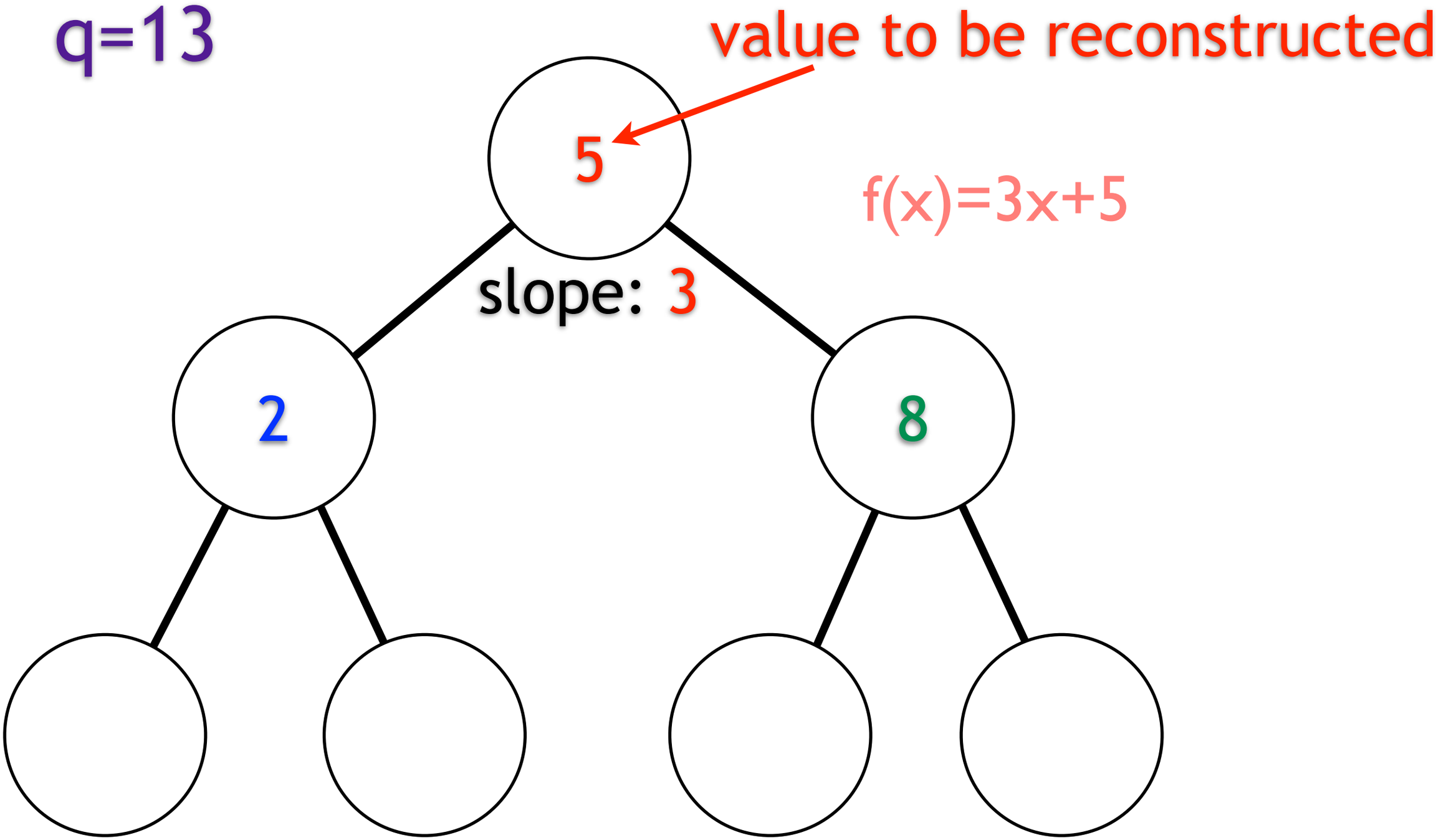


Making the Shares

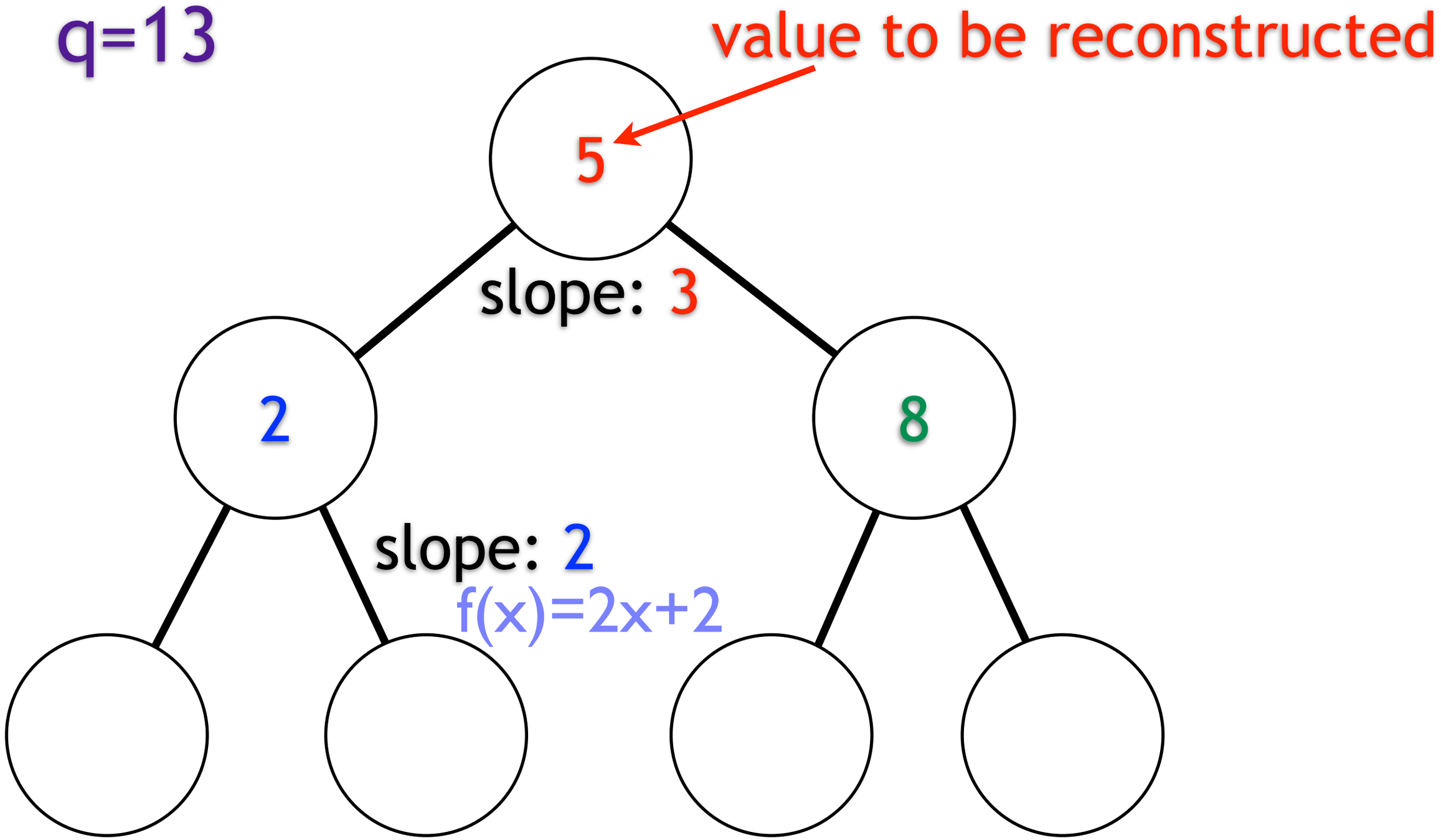
q=13



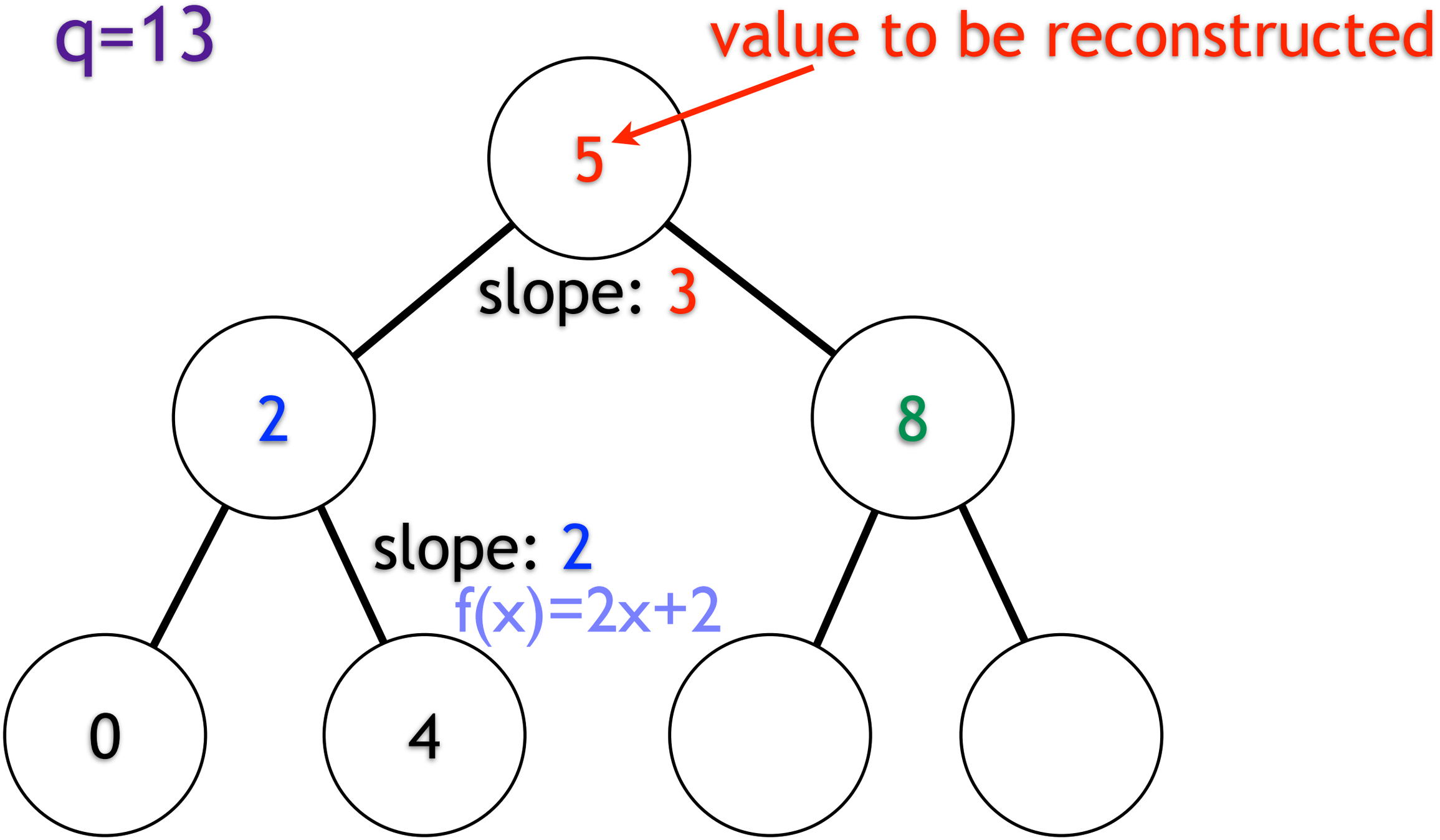
Making the Shares



Making the Shares



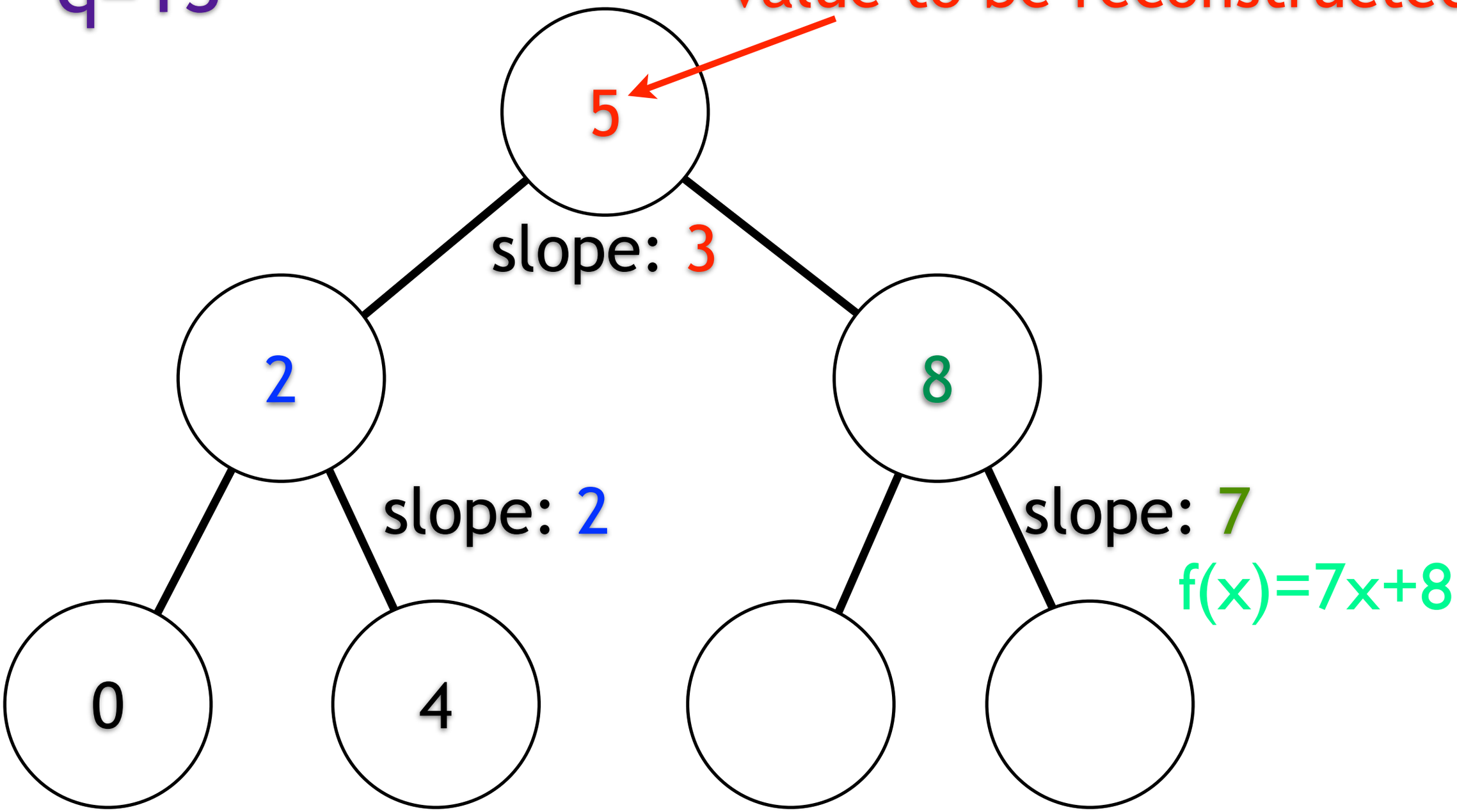
Making the Shares



Making the Shares

q=13

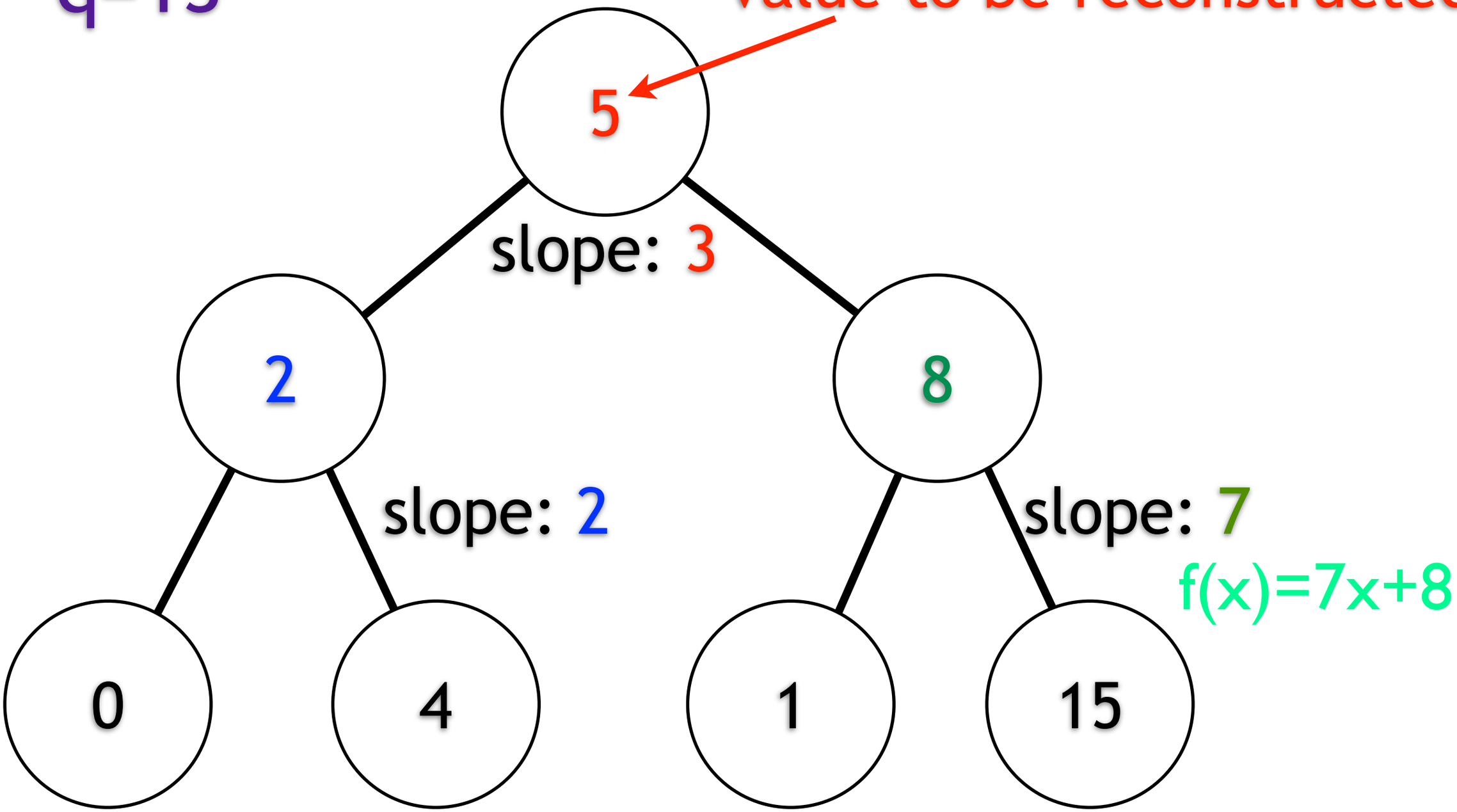
value to be reconstructed



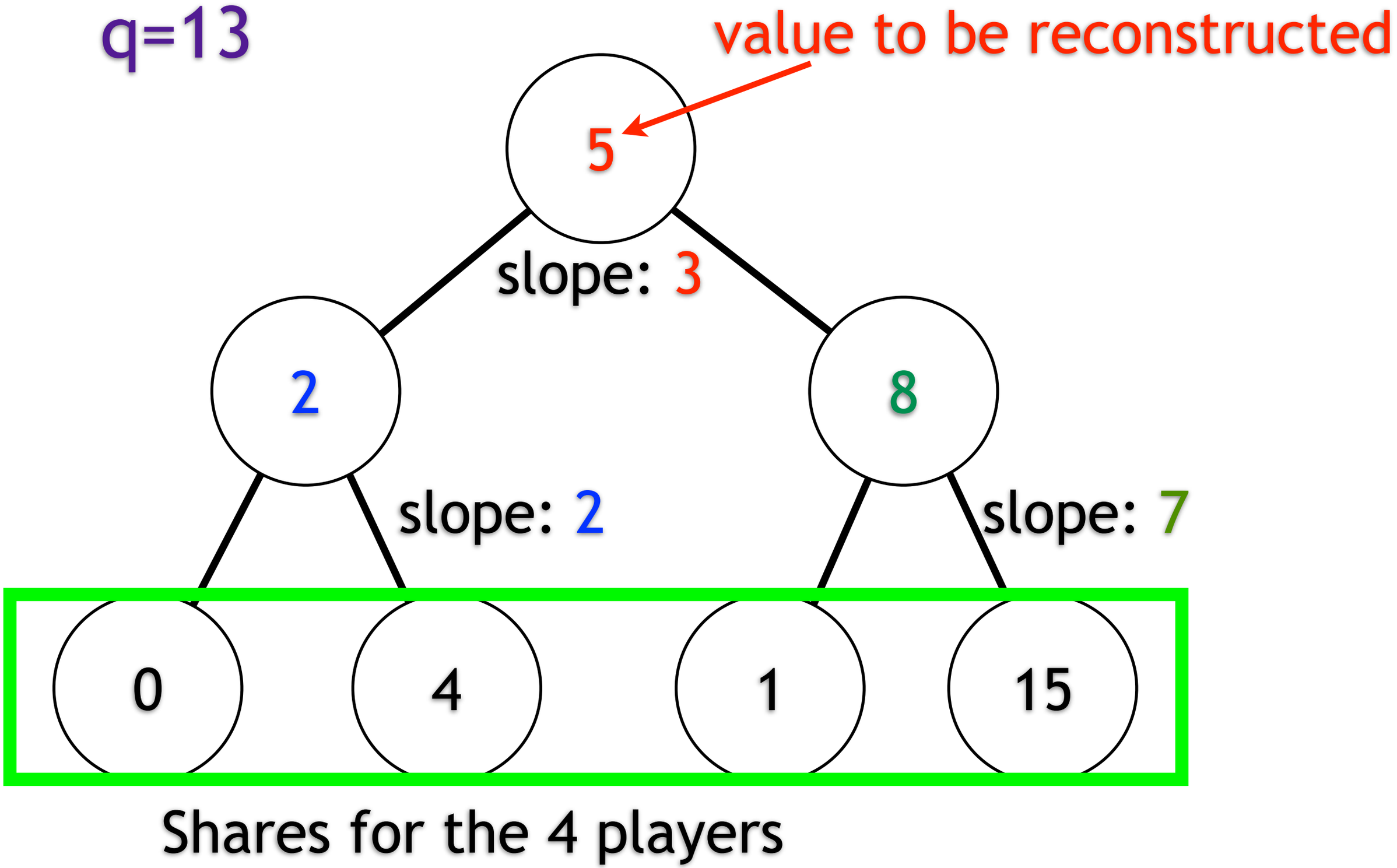
Making the Shares

q=13

value to be reconstructed



Making the Shares



Players' Protocol

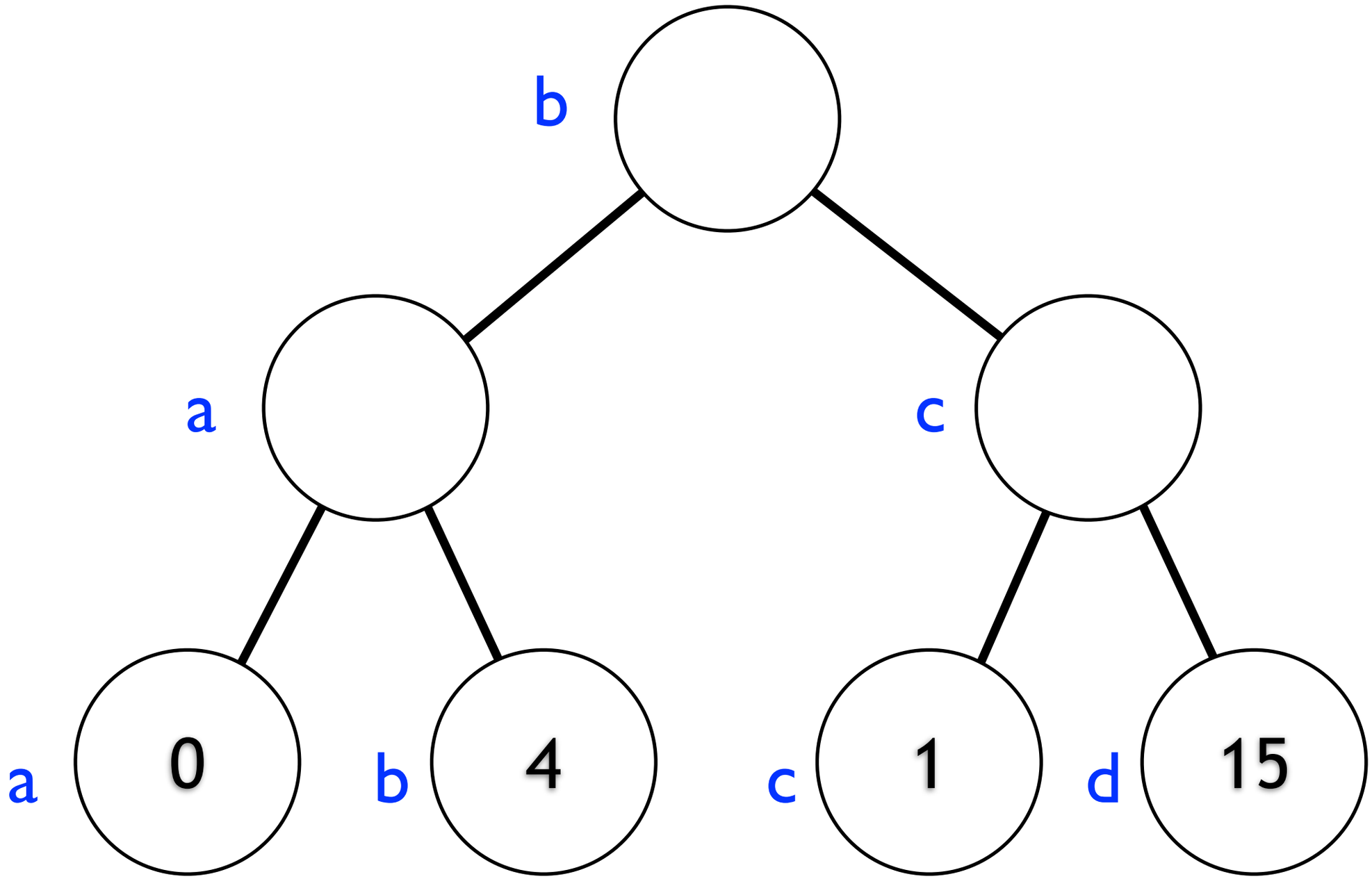
- Two stages per round
  - Up stage: values reconstructed recursively up the tree
  - Down stage: values sent down from root
- All messages authenticated with tag and hashes

Players' Protocol

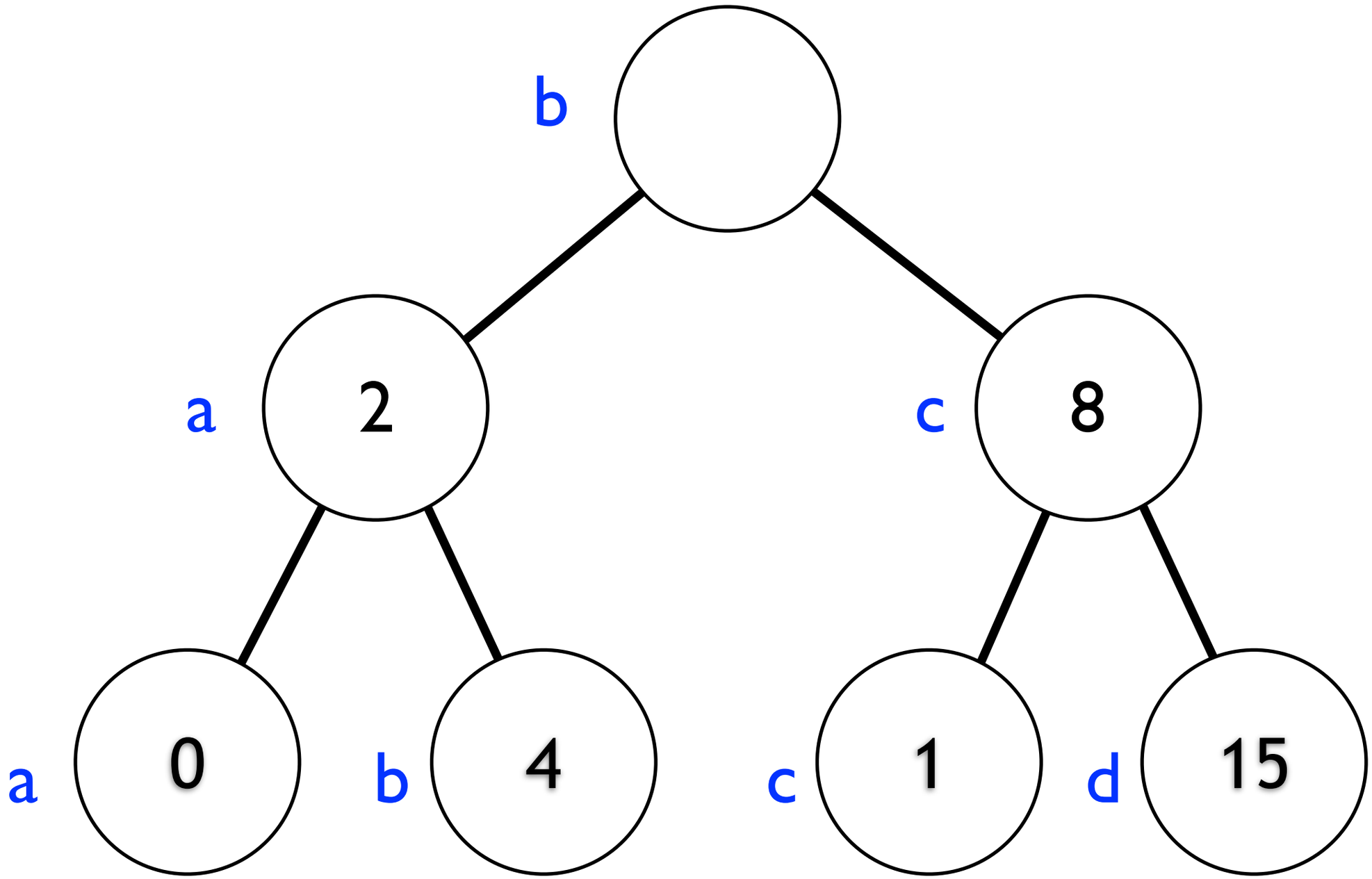
- Two stages per round
  - Up stage: values reconstructed recursively up the tree
  - Down stage: values sent down from root
- All messages authenticated with tag and hashes

If ever detect cheating, output potential secret from current round and QUIT.

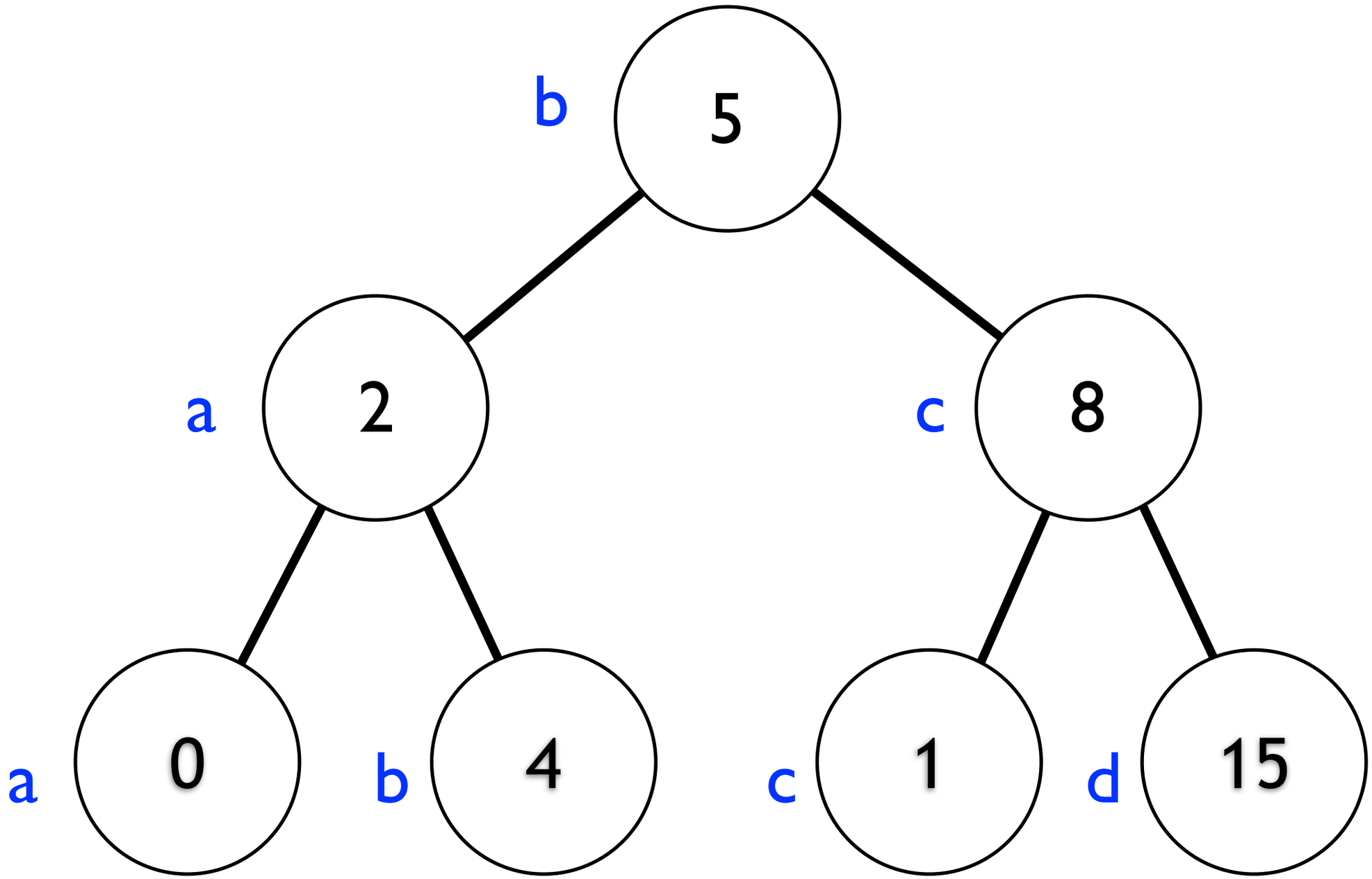
Up Stage



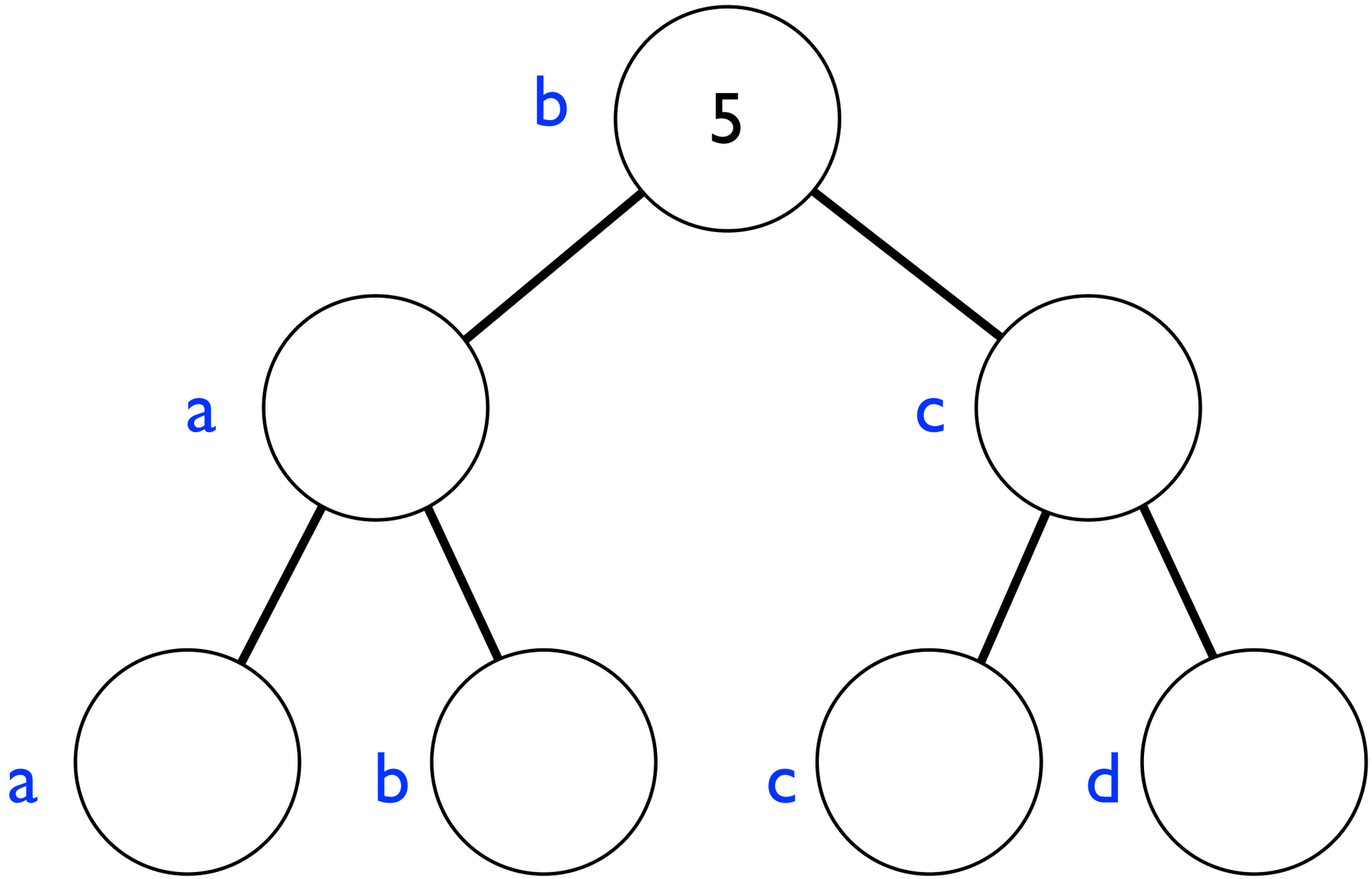
Up Stage



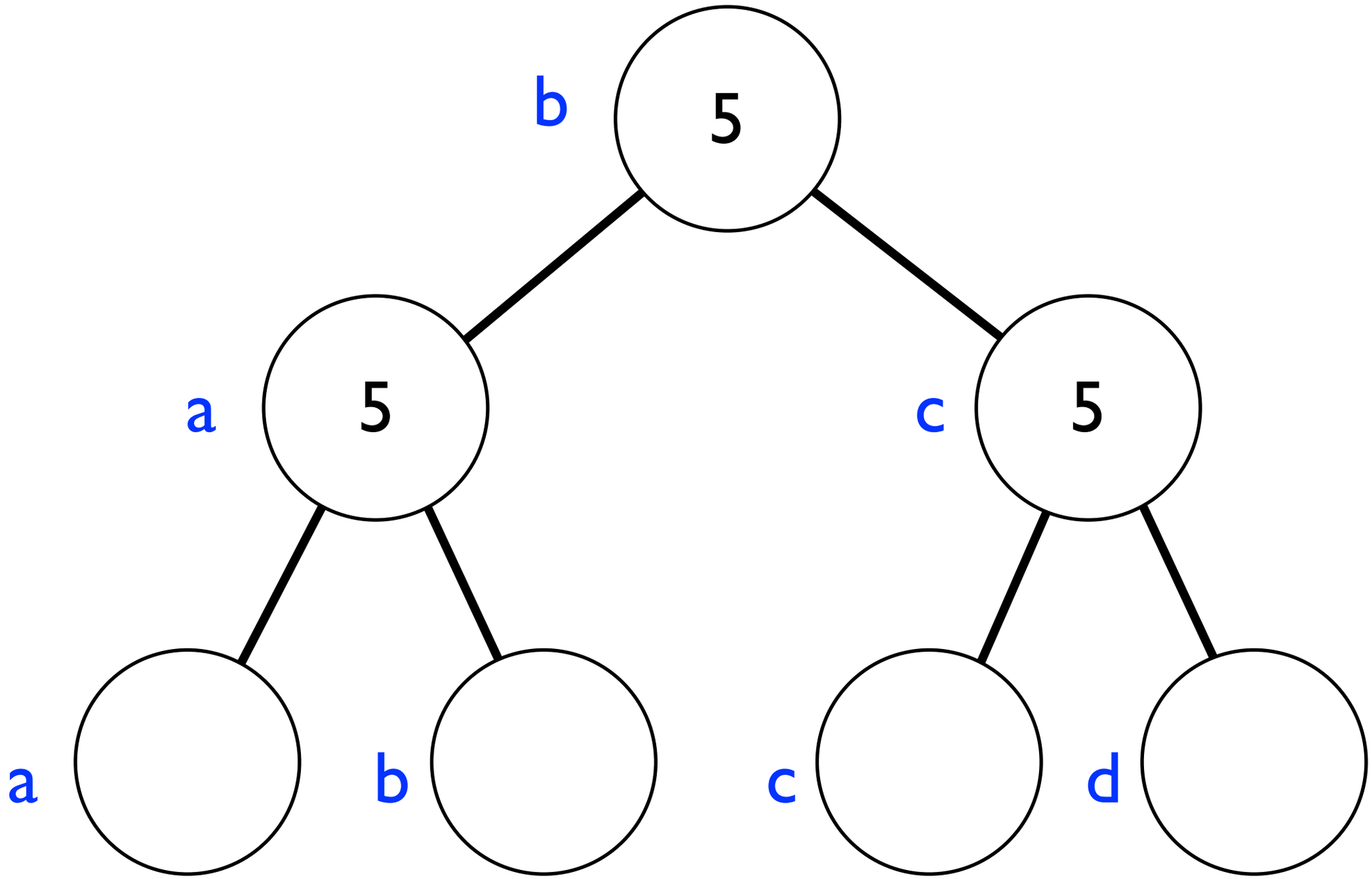
Up Stage



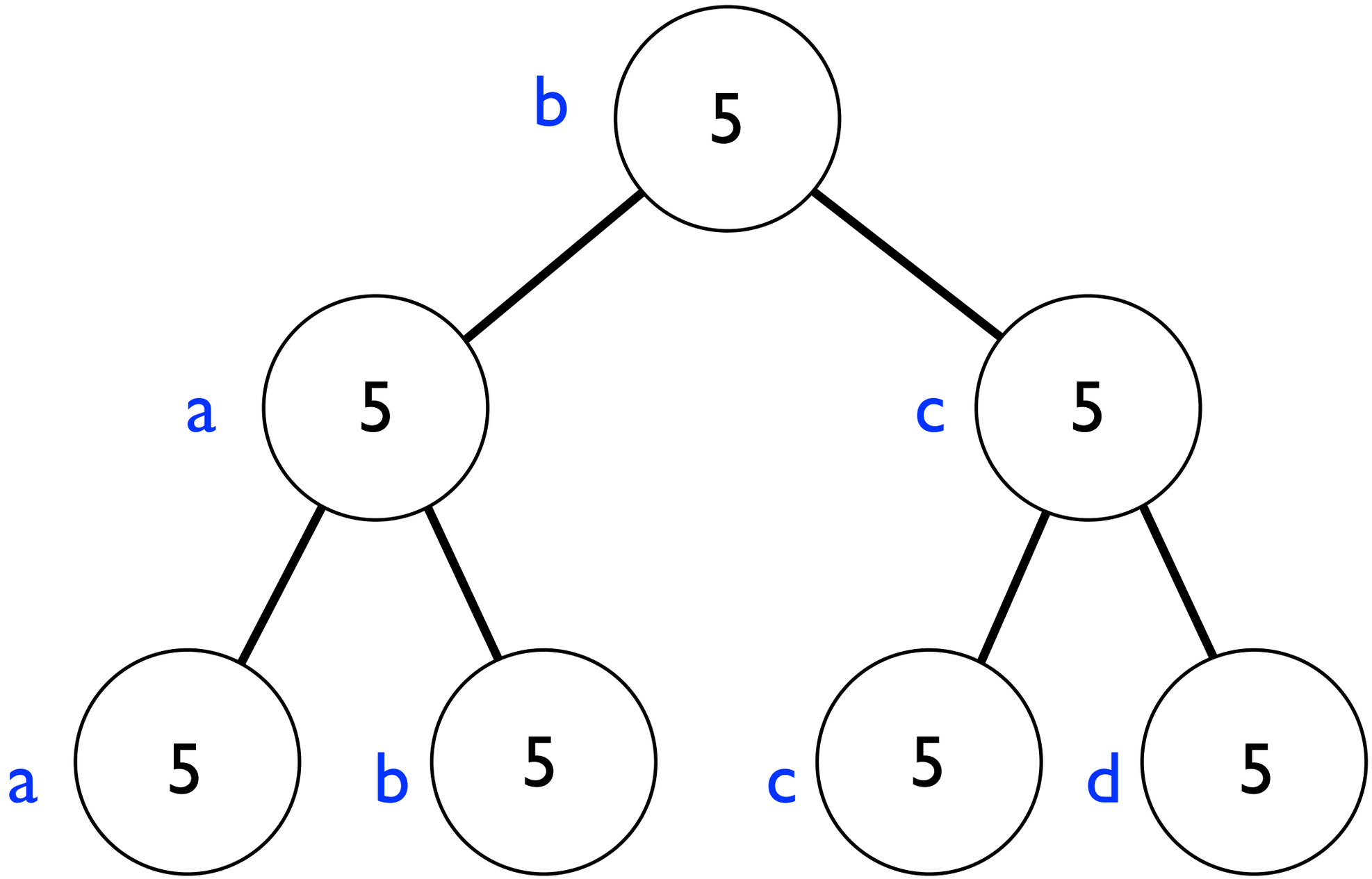
Down Stage



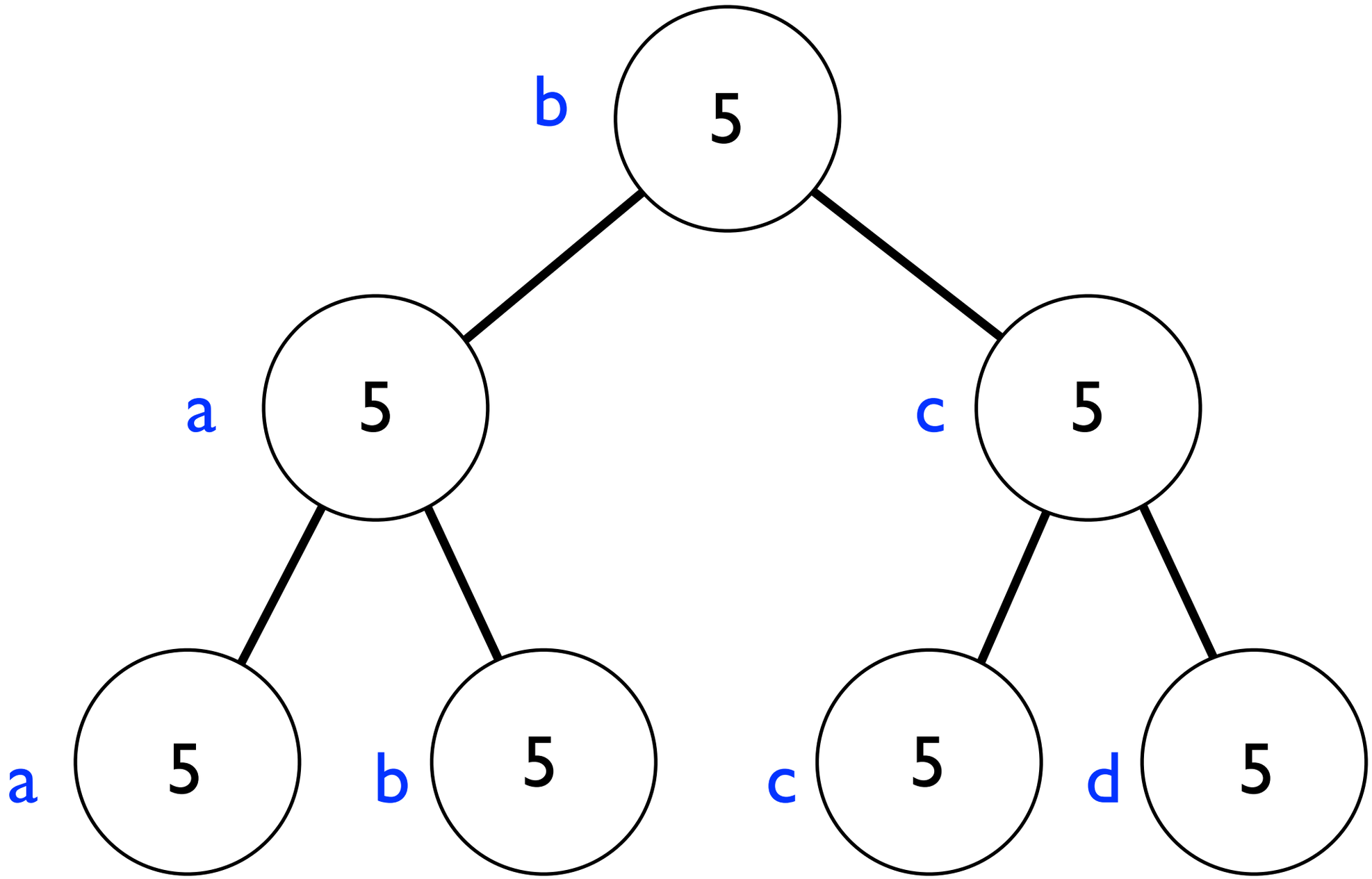
Down Stage



Down Stage



Down Stage



Round

- In each round, reconstruct:
  - Mask for the next round
  - Indicator for whether the current round holds the secret

Termination: At end of down stage, if indicator value is 0, use mask from previous round to decode secret and quit

Why does it work?

- Tag & hash ⇒ can't fake messages.
- Masks ⇒ can't decode secret before round X-1
- No point cheating on round X : everyone outputs secret anyway
- Only viable time to cheat is round X-1 (if you can guess it). Tune parameter β, so that (unless Y is too small), cannot gain more than ε by cheating.

Resource Cost

	Our algorithm	[KN] adaptation
messages (per player-round)	$O(1)$	$\Theta(n)$
latency (per round)	$O(\log n)$	$\Theta(n)$
E(# rounds)	$O(1)$	$\Omega(n)$

Resource Cost

	Our algorithm	[KN] adaptation
messages (per player-round)	$O(1)$	$\Theta(n)$
latency (per round)	$O(\log n)$	$\Theta(n)$
E(# rounds)	$O(1)$	$\Omega(n)$

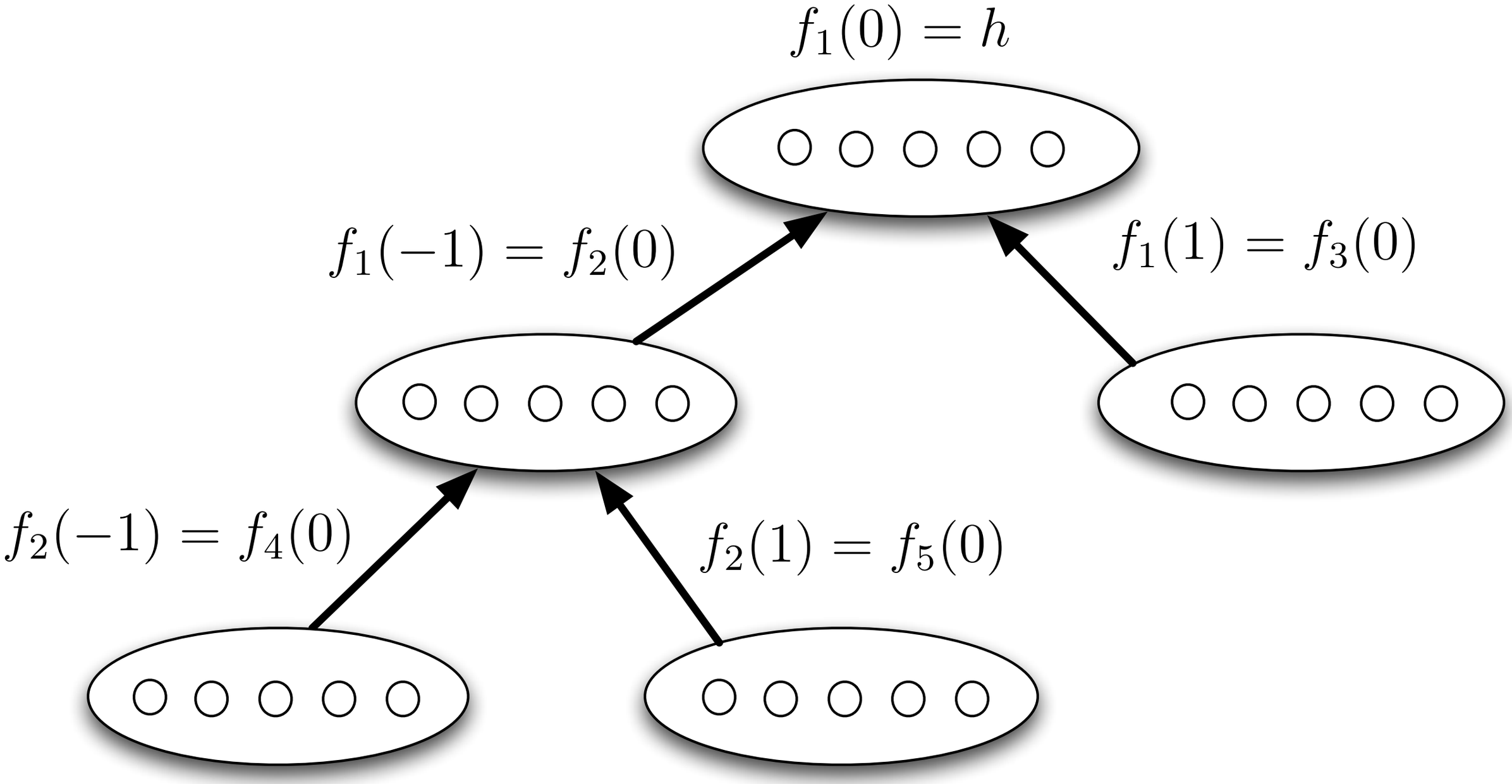
Since β doesn't depend on n

What about m-out-of-n?

- This is difficult in general. If each player sends O (log n) messages, then if m=o(n/log n), even if the active players are chosen randomly, it is likely that there is an active player who never receives a message from another active player.
- Moreover even with Θ(n) active players, if they are selected adversarially, then a small subset of them can always be isolated from the others.
- Need further relaxation of problem

m-out-of-n

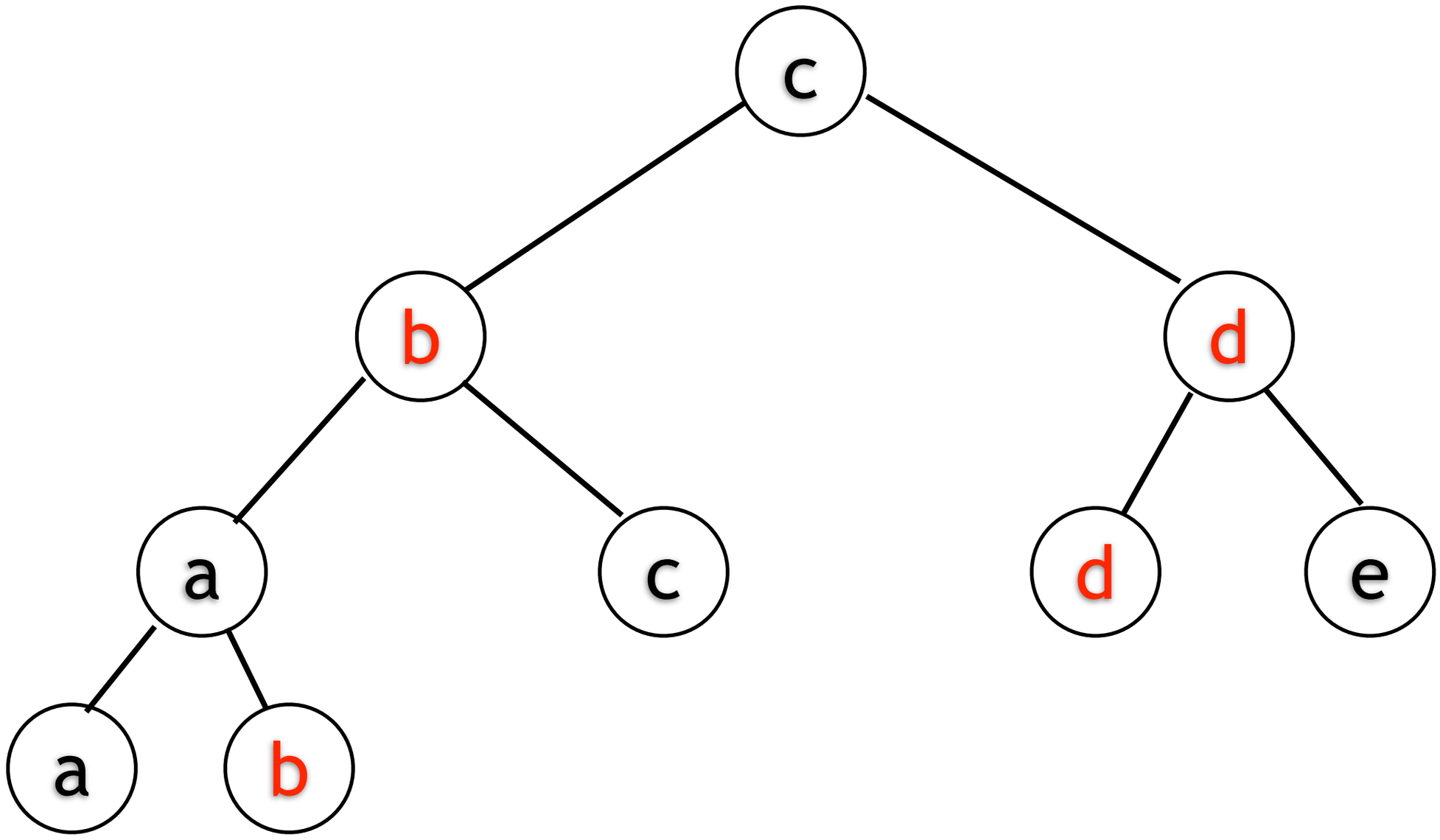
- Suppose m = Θ(n).
- Moreover, suppose the subset of active players is independent of the dealer's coins
- Threshold fraction τ, parameter λ, and an algorithm such that cooperating is ε-Nash, and
  - if m >(τ+λ)n then whp all active players learn
  - if m <(τ-λ)n then whp secret not reconstructed


$f_1(0) = h$
$f_1(-1) = f_2(0)$
$f_1(1) = f_3(0)$
$f_2(-1) = f_4(0)$
$f_2(1) = f_5(0)$

Reducing Error

- X and Y i.i.d. ~ Geom(β)
  X+1: position of secret
  Y: padding on long input
- Half the players are now short (length X)
- Last round: Long players have the secret; Short players know this is last round
- Must carefully distribute short players in last round so they're not descendants of each other

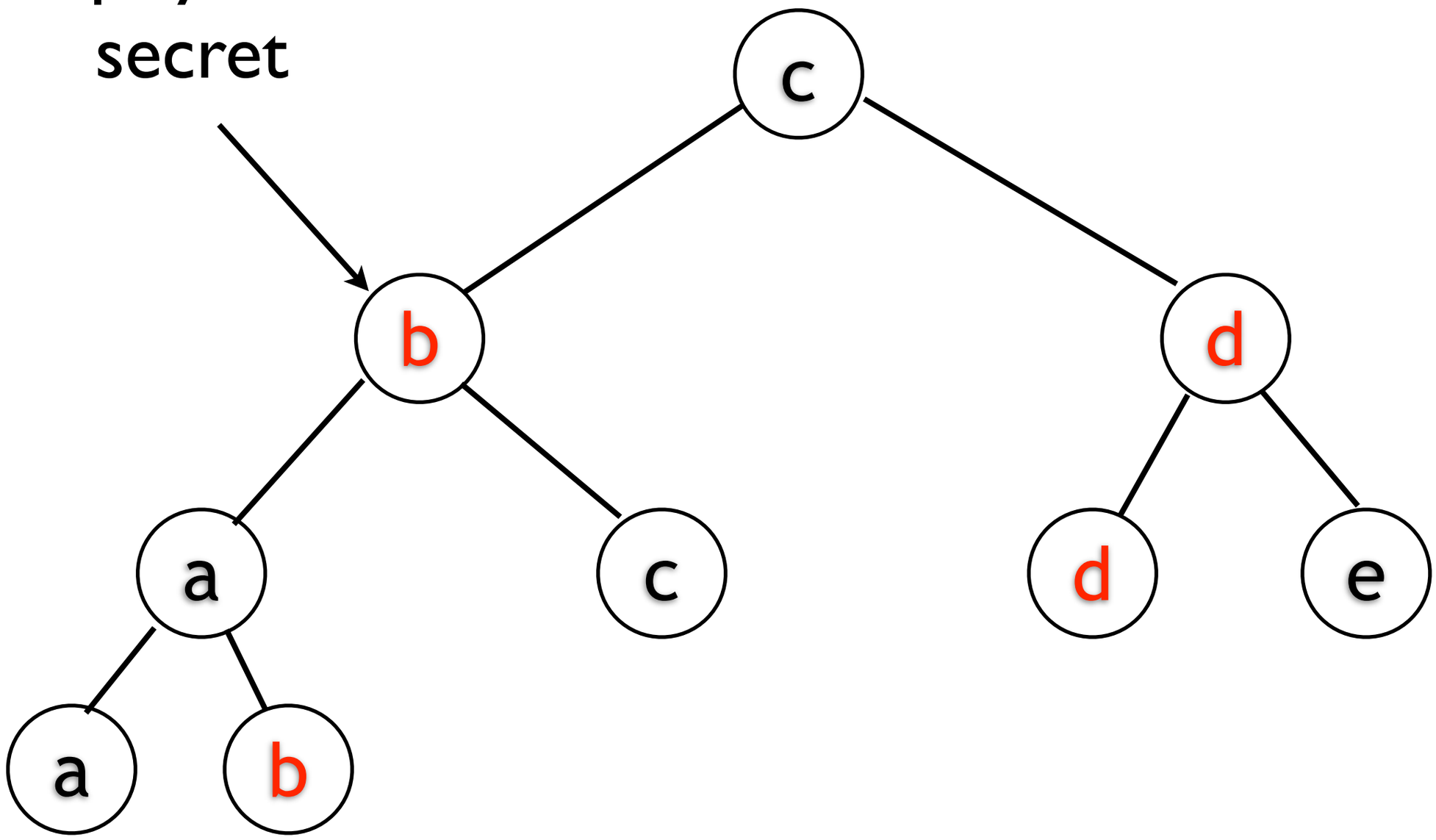
Last Round



red = short

Last Round

short player learns secret

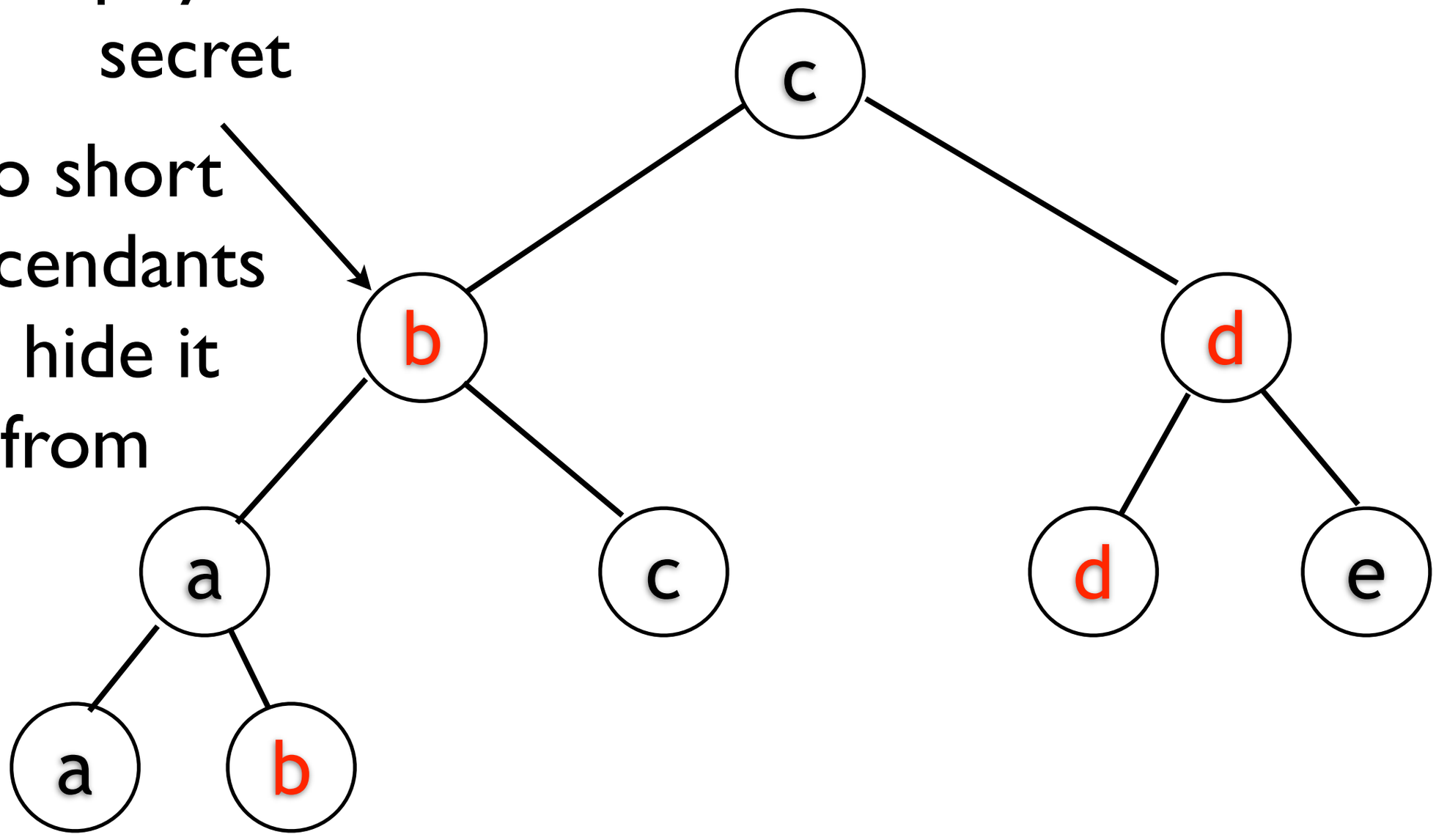


red = short

Last Round

short player learns secret

No short descendants to hide it from



red = short

Probabilities

Size of player i's list

$$Pr(X = t+1 | X \geq t \wedge s_i = k)$$

$$= \frac{Pr(X = t+1 \wedge s_i = k)}{Pr(X \geq t \wedge s_i = k)}$$

$$= \frac{Pr(\bar{\xi}_i \wedge X = t+1 \wedge s_i = k)}{Pr(\xi_i \wedge X \geq t \wedge s_i = k) + Pr(\bar{\xi}_i \wedge X \geq t \wedge s_i = k)}$$

$= Pr(\xi_i)\beta(1-\beta)^k$ $\qquad$ $Pr(\bar{\xi}^i)\beta^2(1-\beta)^{k-1}(k-t+2)$

$Pr(\bar{\xi}_i)\beta^2(1-\beta)^{k-1}$

Conclusions

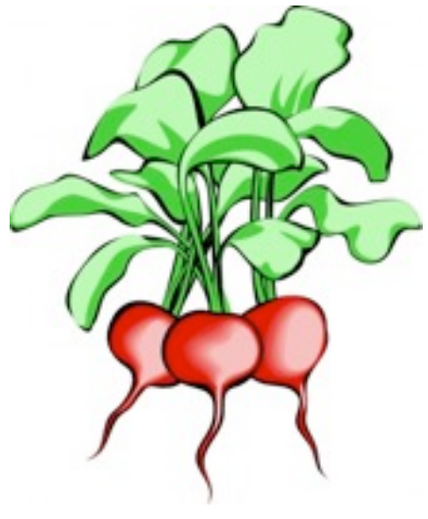
- n-out-of-n rational secret sharing using polylog resources
- Extends to m-out-of-n for $m=\Theta(n)$ randomly chosen active players

Open Problems

- Scalable Rational Secure Multiparty Computation
- Scalable ~~Rational~~ Secret Sharing
  - Better results for m-out-of-n
    - for m = o(n), suitably easier problem must be considered.
    - For m = Θ(n), is anything at all possible for worst-case choice of active players?

Other Applications

- Run an auction or hold a lottery to allocate resources in a network
- Beet Auctions (double auction to determine market clearing prices for sugar beets between Danish beet farmers and the sugar company.)



Communication

- Naive: Simulate broadcast on private channels by sending same message to everybody
  - Problem 1: With channels, C doesn't know what A said to B
  - Problem 2: Requires Θ(n) messages per player