GT Algorithmique Distribuée - 7 Mars 2016

TIME VS. INFORMATION TRADEOFFS FOR LEADER ELECTION IN ANONYMOUS TREES

Christian Glacet

CNR - IEIIT, Turin, Italie

Avery Miller

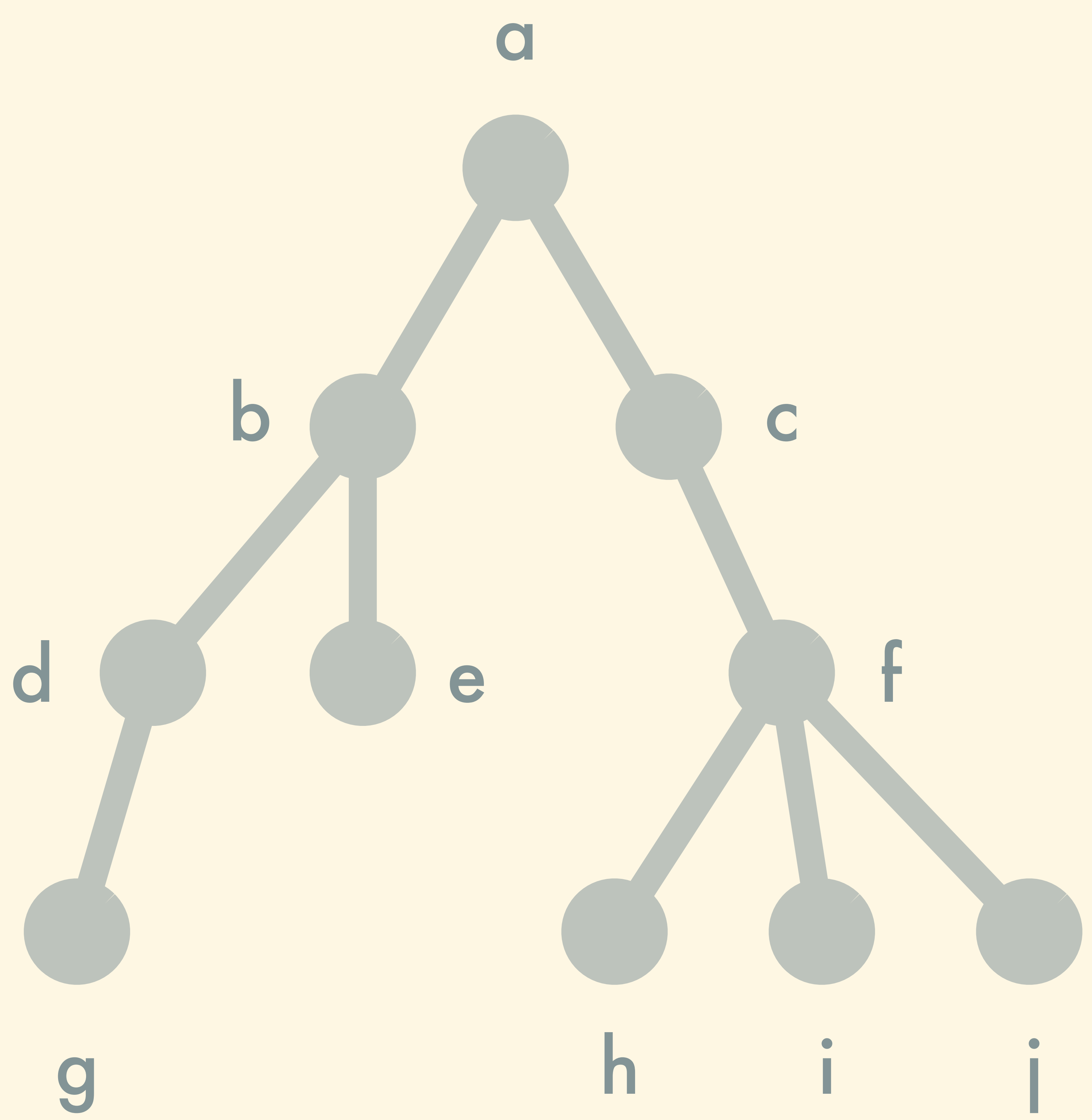
Université de Tel-Aviv, Israël

Andrzej Pelc

UQO, Gatineau, Canada

Leader election problem

Anonymous election ?

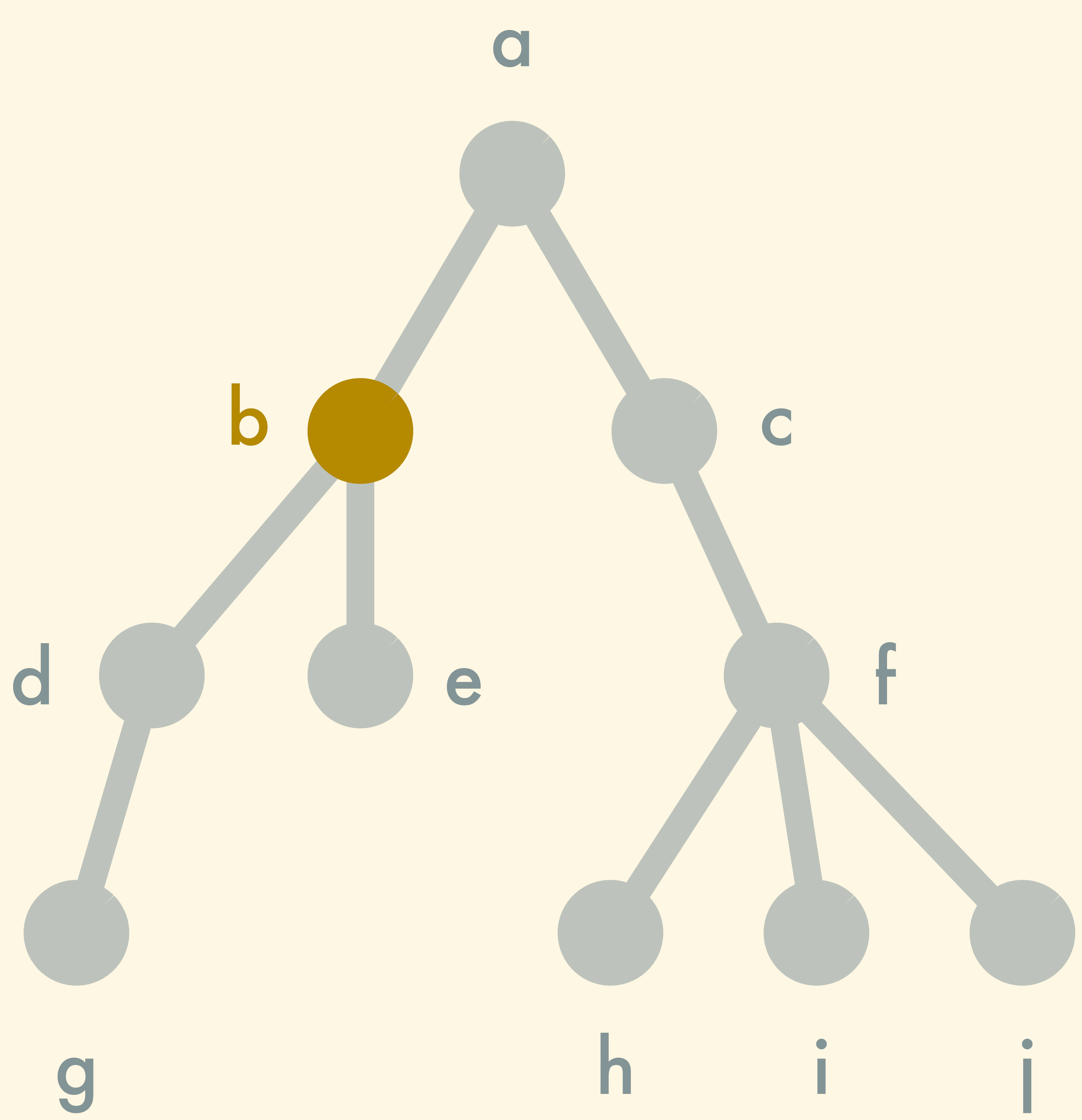


INPUT

OUTPUT

Leader election problem

Anonymous election ?



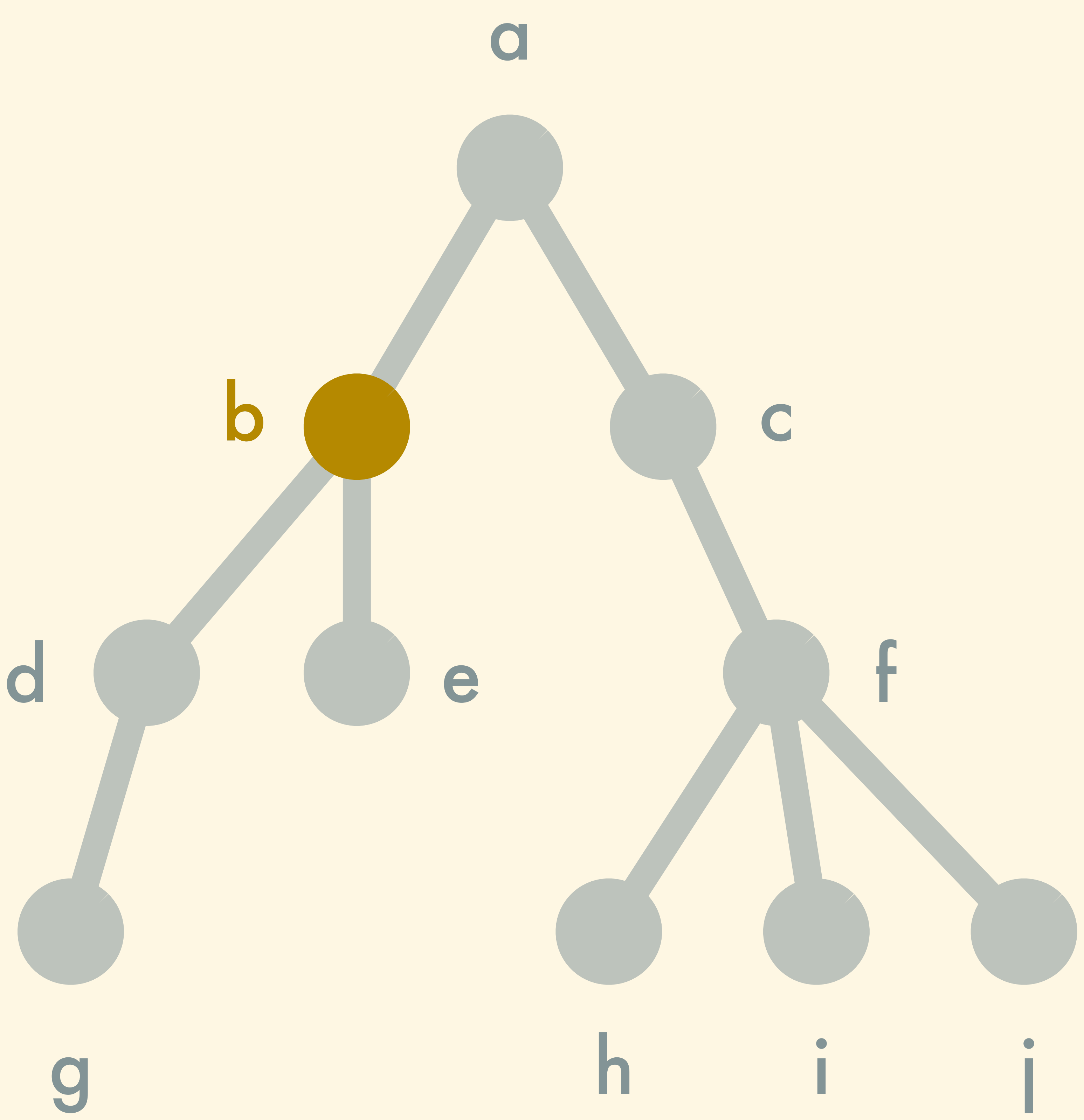
INPUT

Agree on a node ID: b

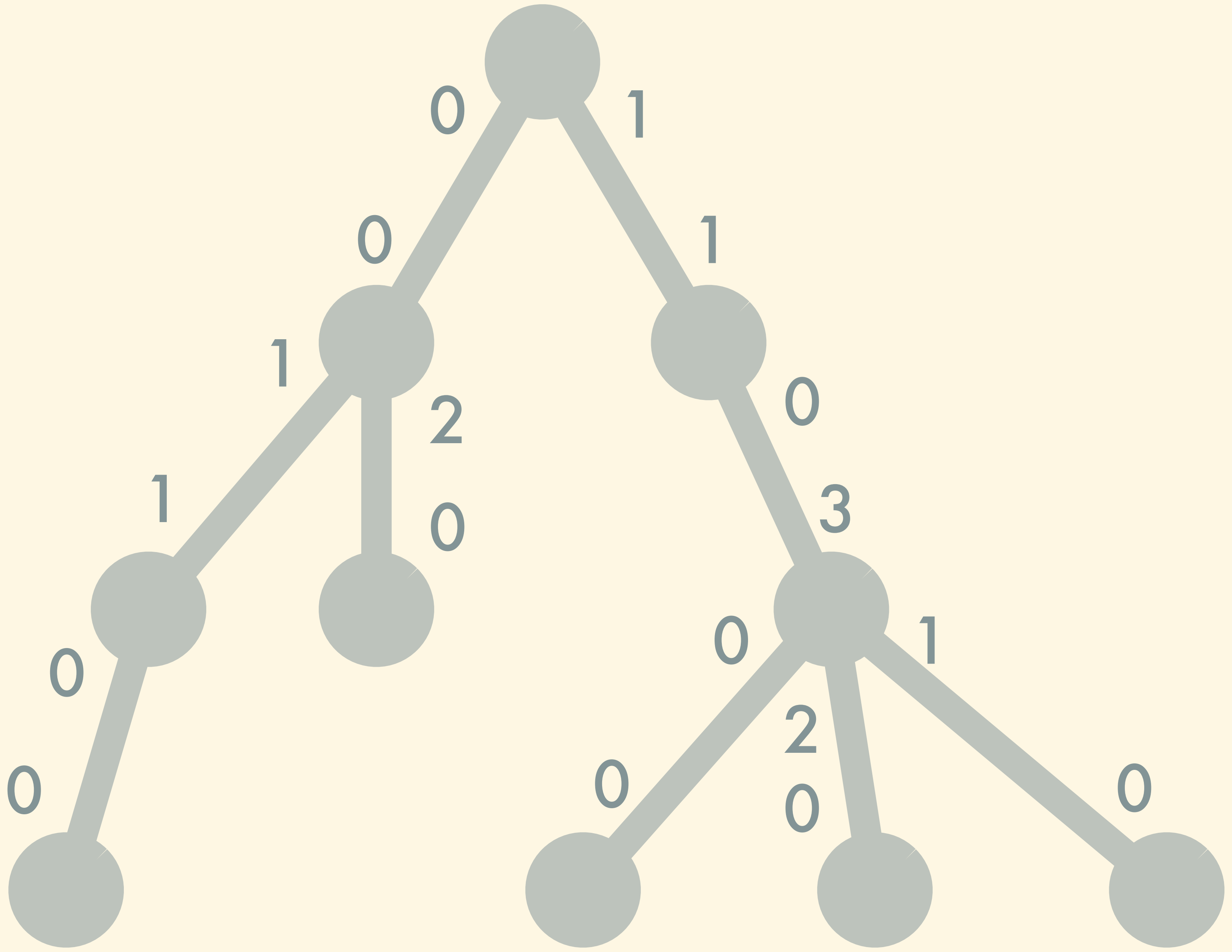
OUTPUT

Leader election problem

Anonymous election ?



INPUT

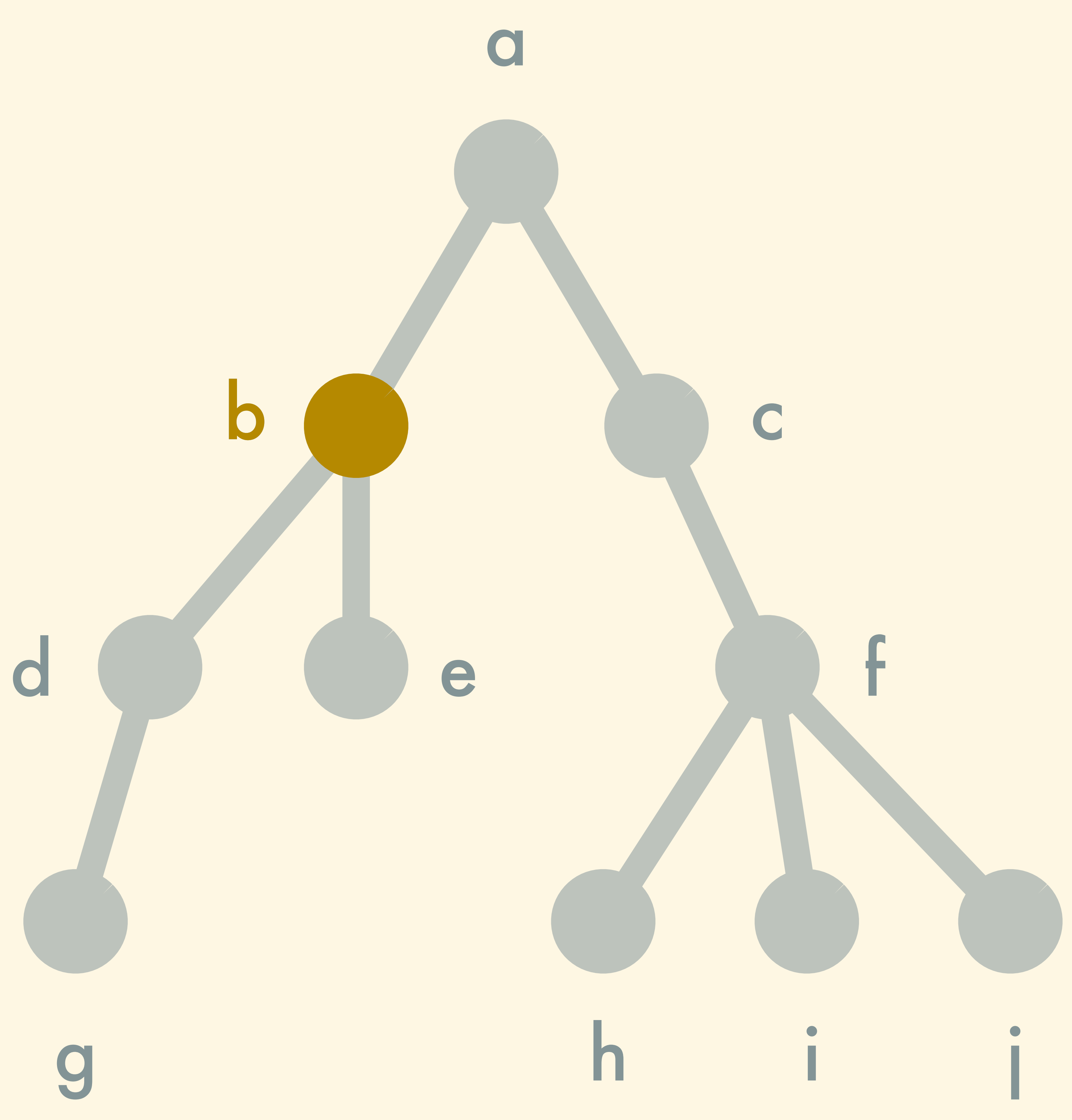


Agree on a node ID: b

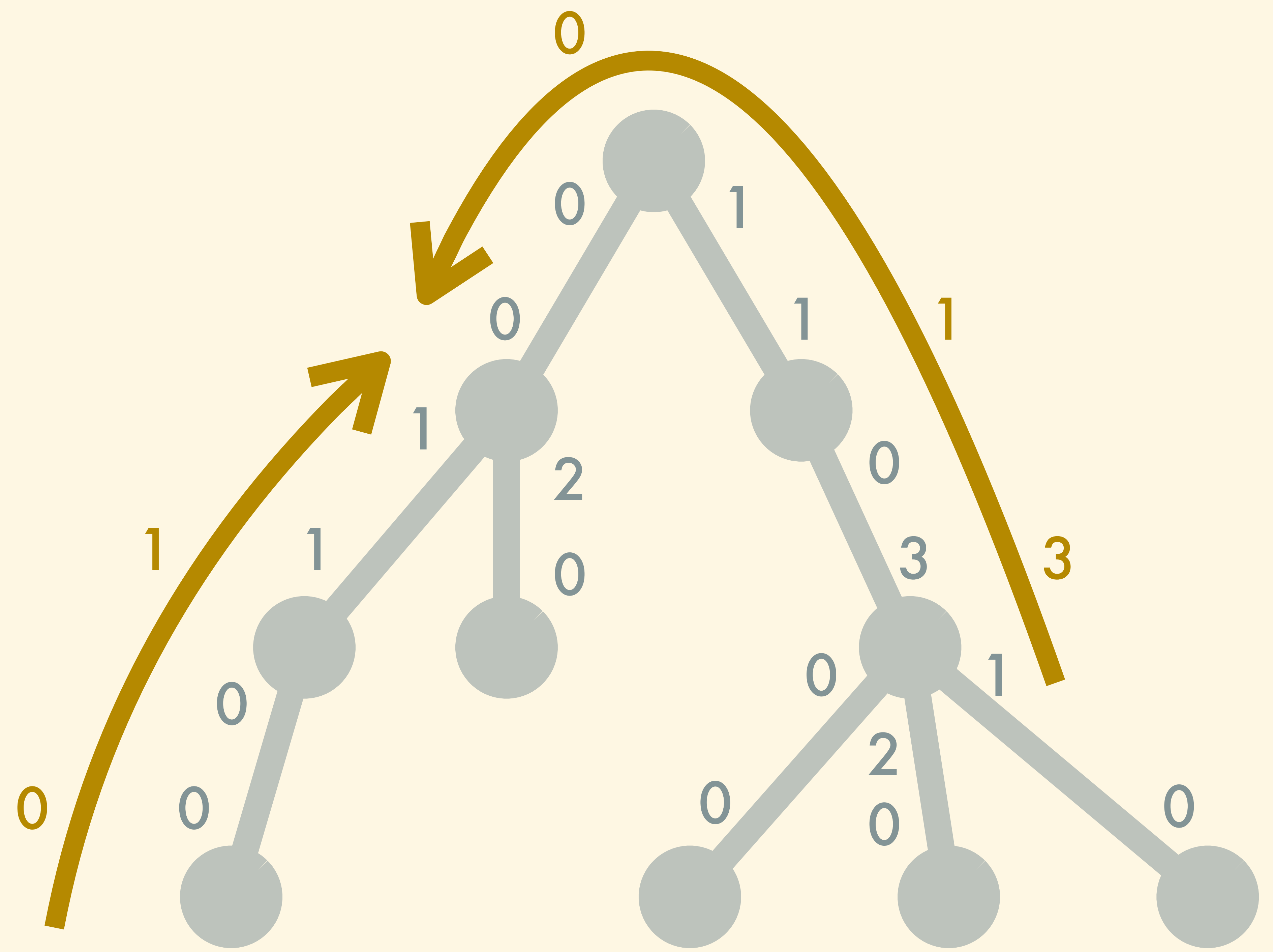
OUTPUT

Leader election problem

Anonymous election ?



INPUT

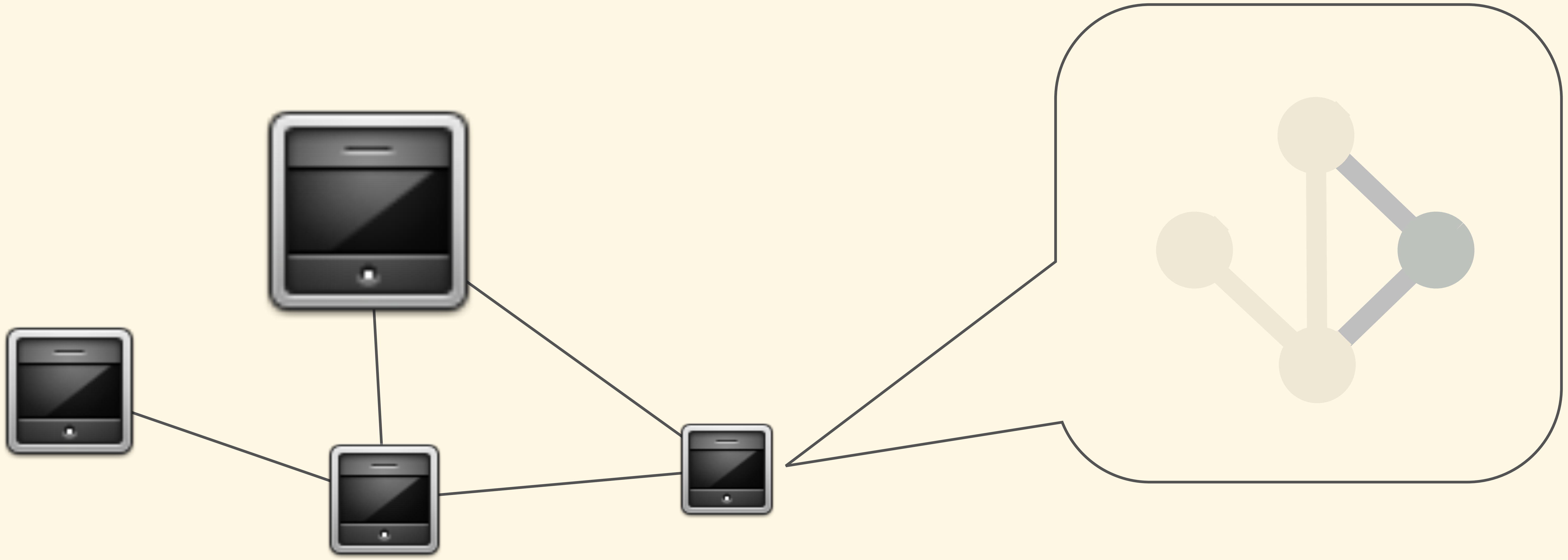


Agree on a node ID: b

OUTPUT

Agree on a node "ID": (0,1) / (3,1,0) /...

Distributed computation



Partial knowledge of the graph (local)

⇒ need of communication to discover the graph

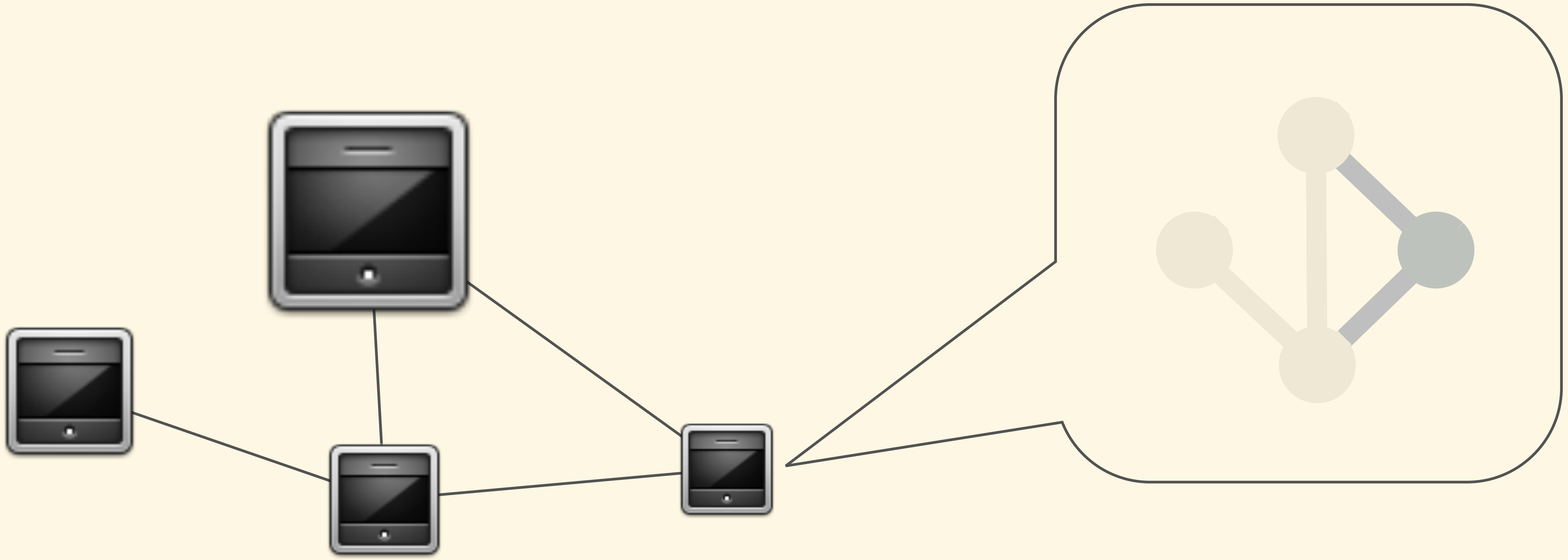
What is typically observed

Time

Memory

Communication cost

Distributed computation



Partial knowledge of the graph (local)

⇒ need of communication to discover the graph

What is typically observed

Time

Memory

Communication cost

Observations

- In trees, election can be done in time D
  - Nodes see the whole graph and can:
    - locate itself ;
    - elect locally ;
    - output the path to the leader ;
- Without any additional info. election is not possible in time < D.
- Helping nodes by giving them: the size of the network n, the diameter D, ...

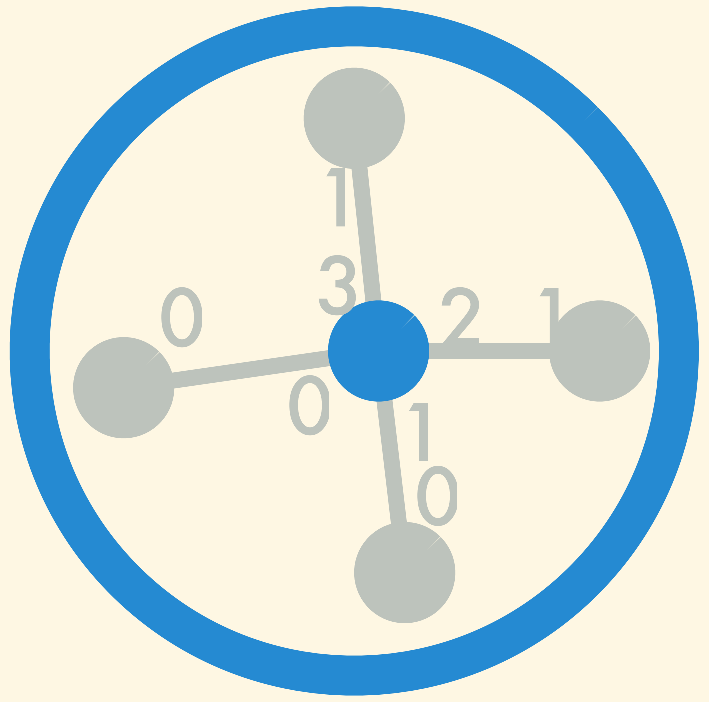
More general question

- What amount of information has to be known so election can be done in time at most τ < D ? (for different values of τ)

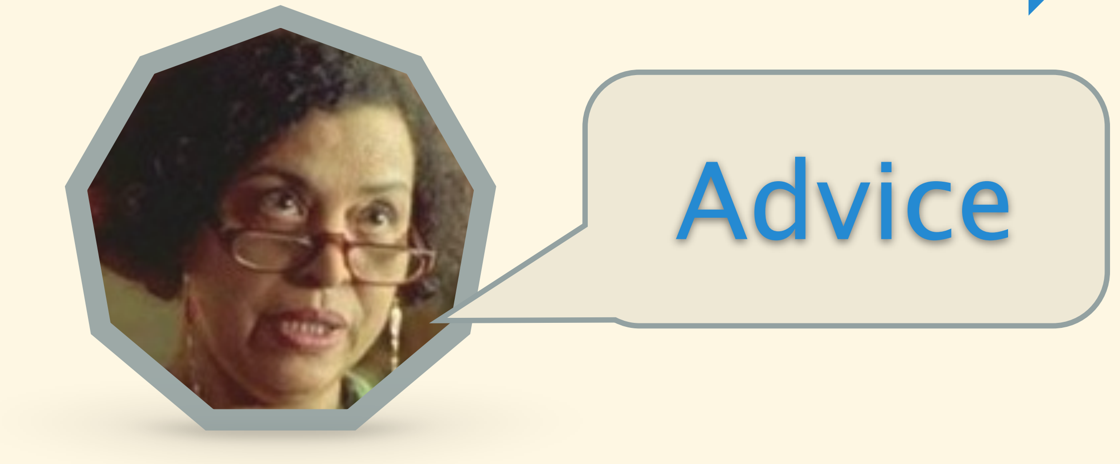
- Remark: we only observe graphs where election is possible in time τ.
    - Intuitively, graphs that are not too symmetric.

- Things we **don't** look at:
    - Communication/memory costs, local computation power, …
    - Randomised algorithms

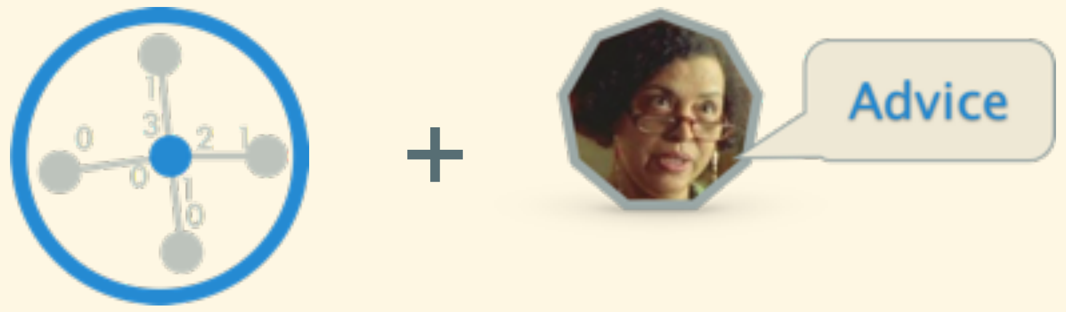
Details of the Model



- Local model (input and output ports).
  - After r rounds, a node knows the ball of radius r



- An oracle gives an advice to all nodes (a single advice is shared).
  - The oracle sees the whole graph
  - We look at the size (in bits) of the advice



- Every node use its local view (ball) and the advice to elect.
  - Election: all nodes output a simple path, all paths lead to the same node.

RESULTS

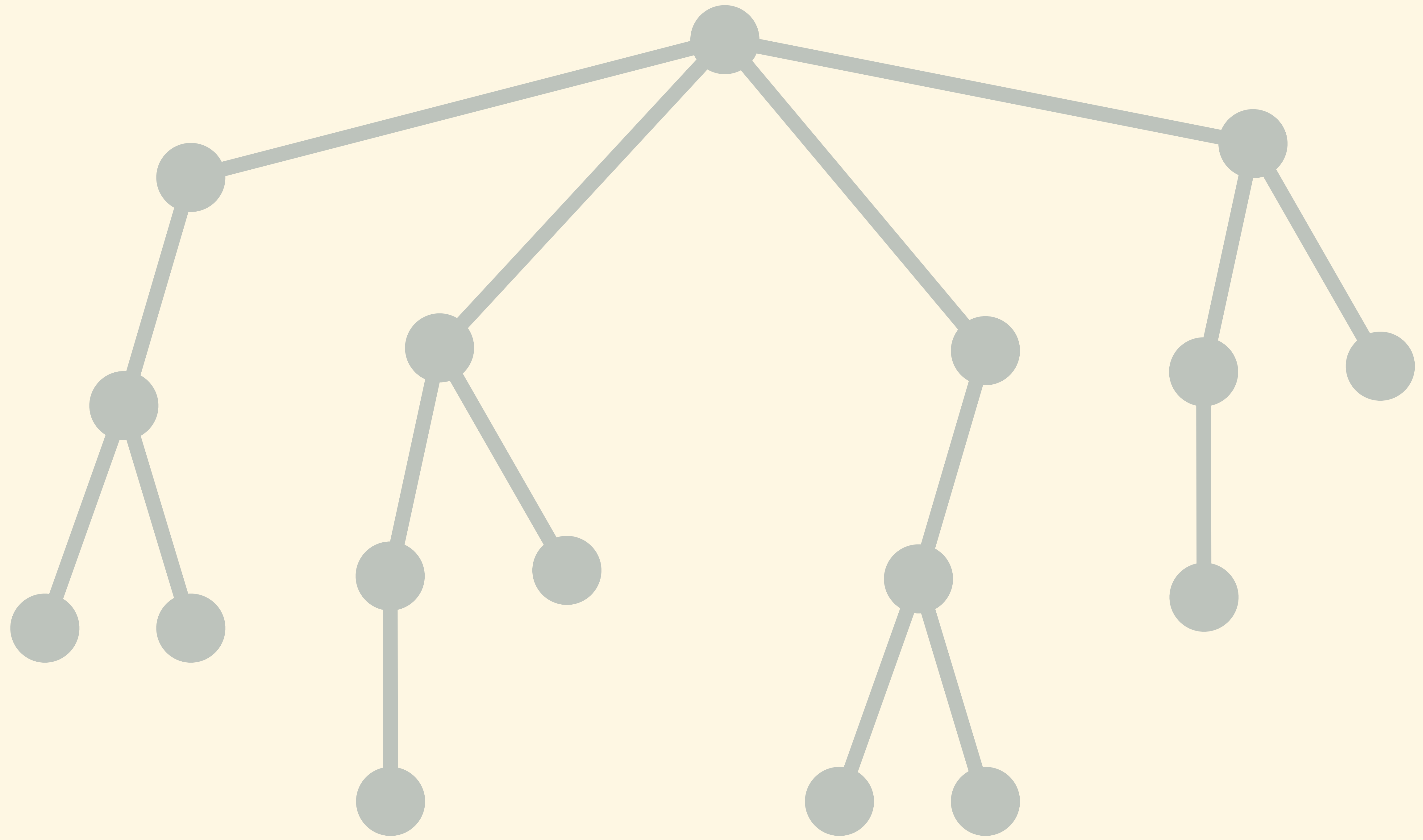
Time	Advice size	Remark
D	0	
D-1	Θ(log D)	
D-2	Θ(log D) Θ(log n)	D even D odd
[β*D, D-3], β > 1/2	O(n log n/D) Ω(n/D)	upper bound D even or τ < D-3
α*D, α < 1/2	Θ(n)	except when D is small (D ∊ o(log n))

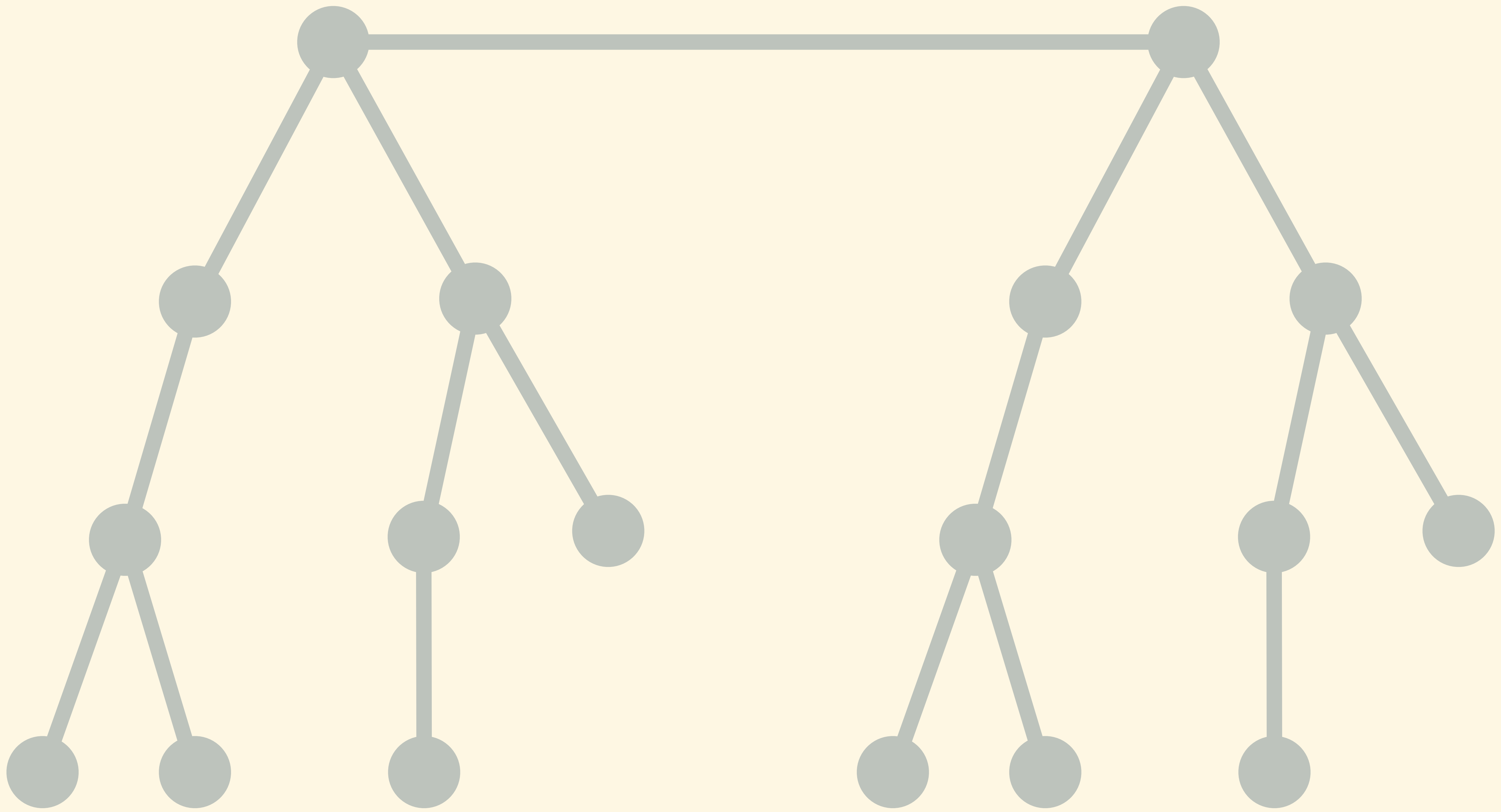
Upper bounds

Key ideas

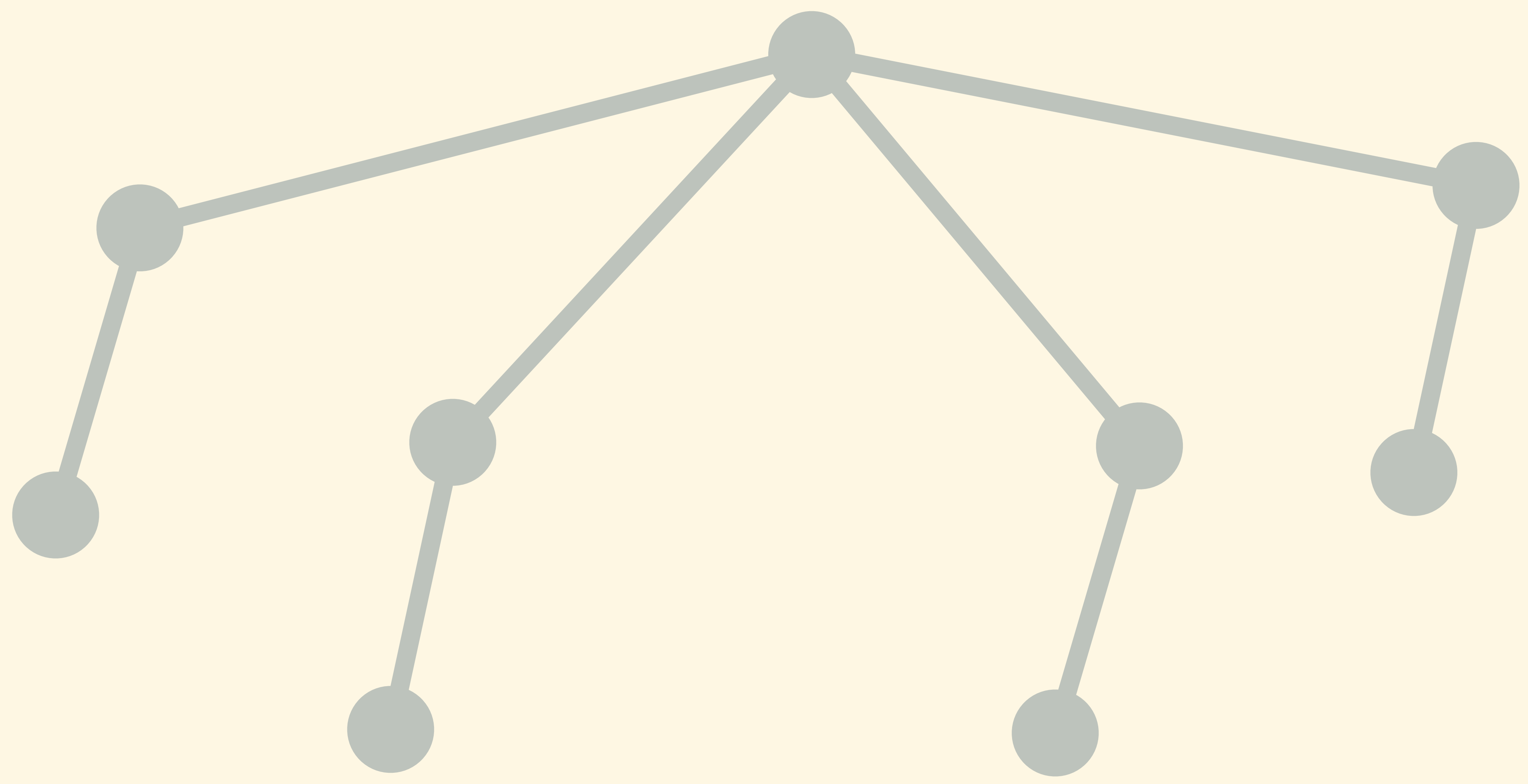
- Elect the around the center:
  - Even diameter: elect the central node
  - Odd diameter
    - find the central edge
    - elect one of its end
- Advice:
  - Find subgraphs that only a portion of the nodes can see
  - Use it to give personalised advices
    - "if you see X then do …"

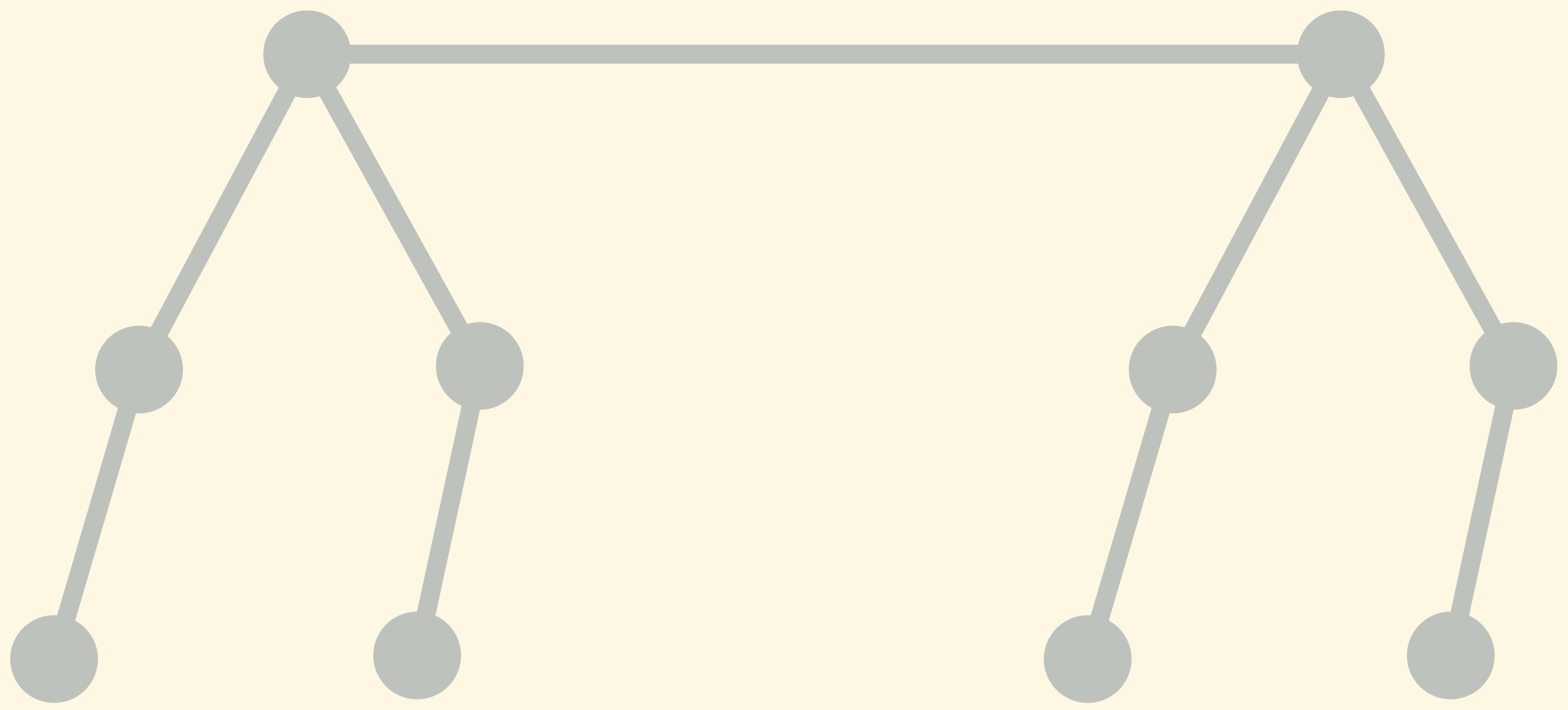
Central node/edge





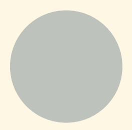
Central node/edge



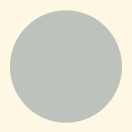


Central node/edge

Central node/edge



CENTRAL NODE/EDGE



Even diameter ⇒ it exists a central node

Odd diameter ⇒ it exists a central edge

Lower bounds

Key ideas

- Key idea for lower bounds (generally speaking):
  - Show a family of trees in which for any two given graphs
    - if the oracle gives the same advice,
    - then election fails in one of the two graphs.
  - Show that this family is “big”.

- Take care, election has to be possible in this family of trees !
  - Technic: Prove that election is feasible when the whole graph is given as an advice.

Suite de la présentation

Time	Advice size	Remark
D	0	
D-1	$\Theta(\log D)$	
D-2	$\Theta(\log D)$ **$\Theta(\log n)$**	D even **D odd**
[β*D, D-3], β > 1/2	$O(n \log n/D)$ $\Omega(n/D)$	upper bound D even or $\tau <$ D-3
α*D, α < 1/2	$\Theta(n)$	except when D is small ($D \in o(\log n)$)

ODD DIAMETER, TIME D-2

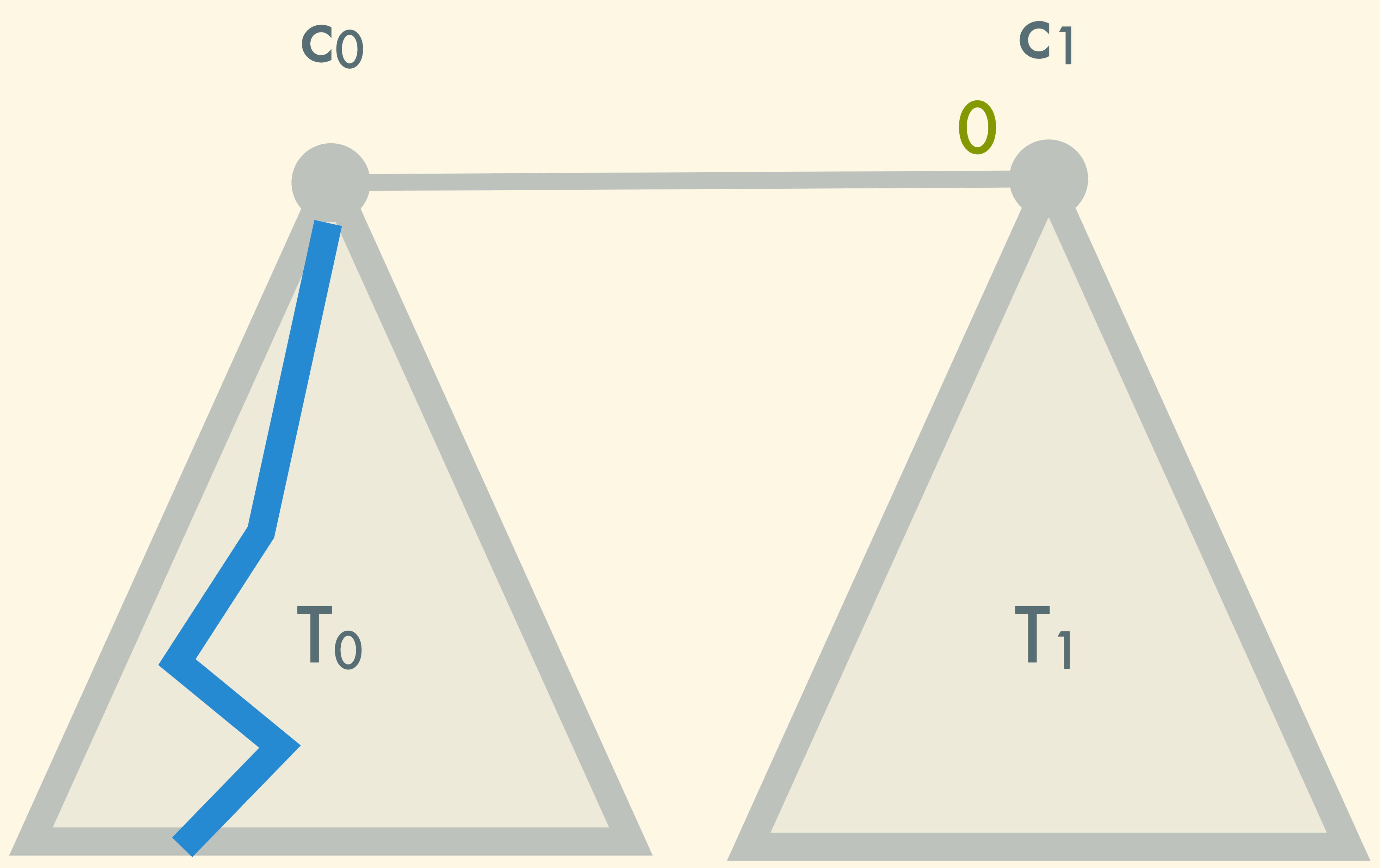
Upper Bound

ADVICE

Advice

- A path that appears in T_0 but not in T_1
- The port number from c_1 to c_0

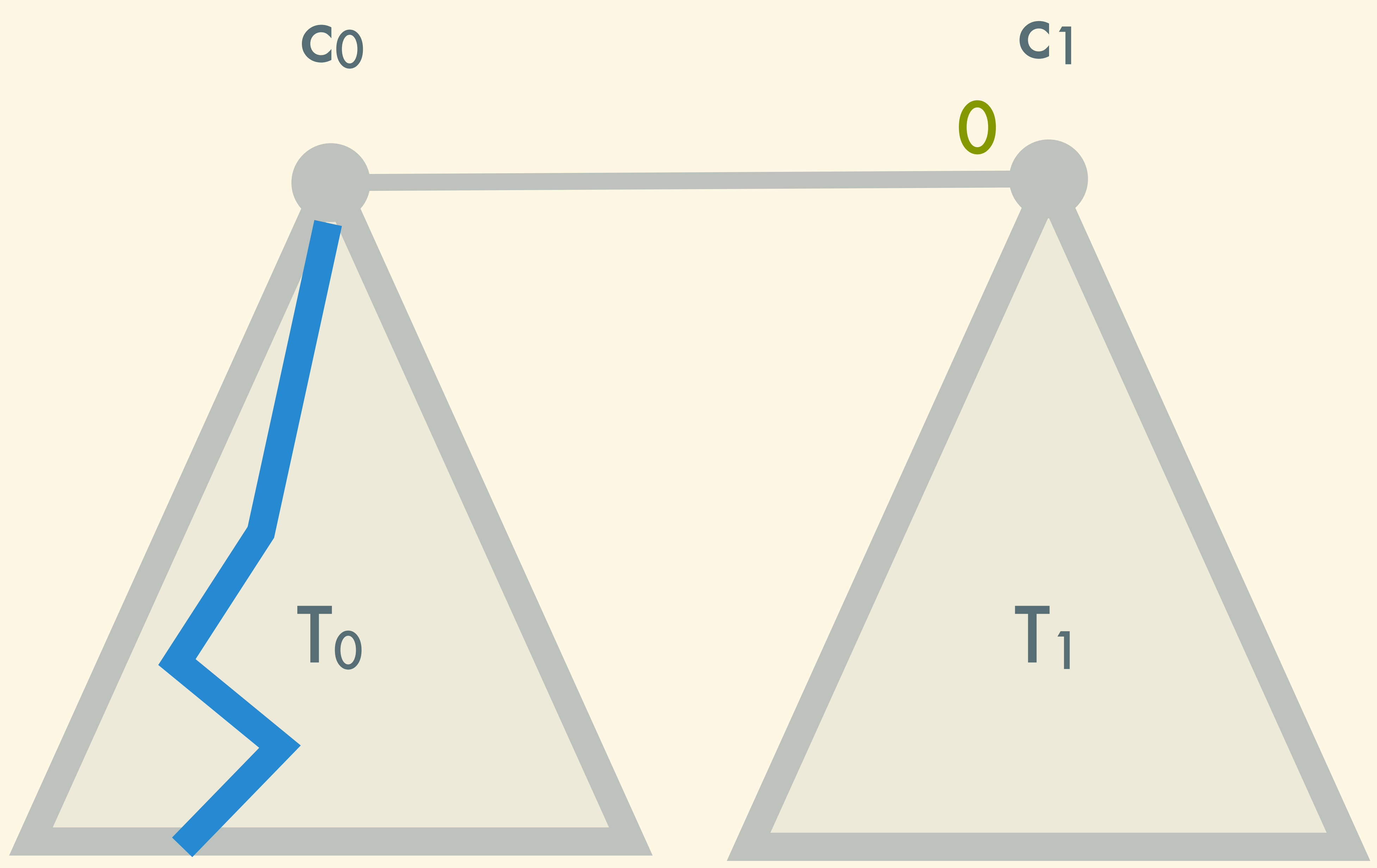
size: $O(D \log n)$



Election

Election for a node u

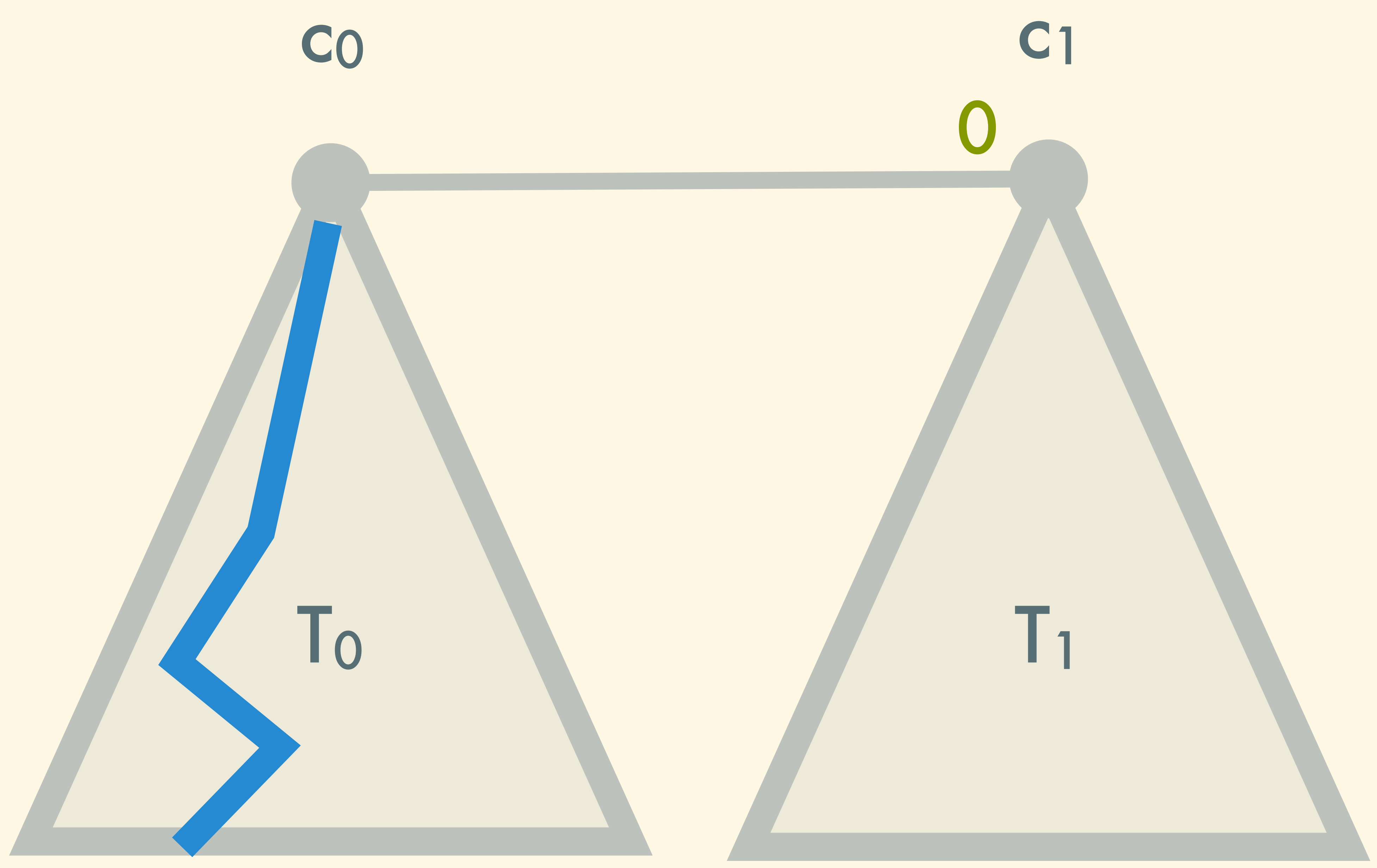
- If a u sees the path
  - then it elects c_0 (how?)
  - otherwise it uses the port number to find/elect c_0



Proof

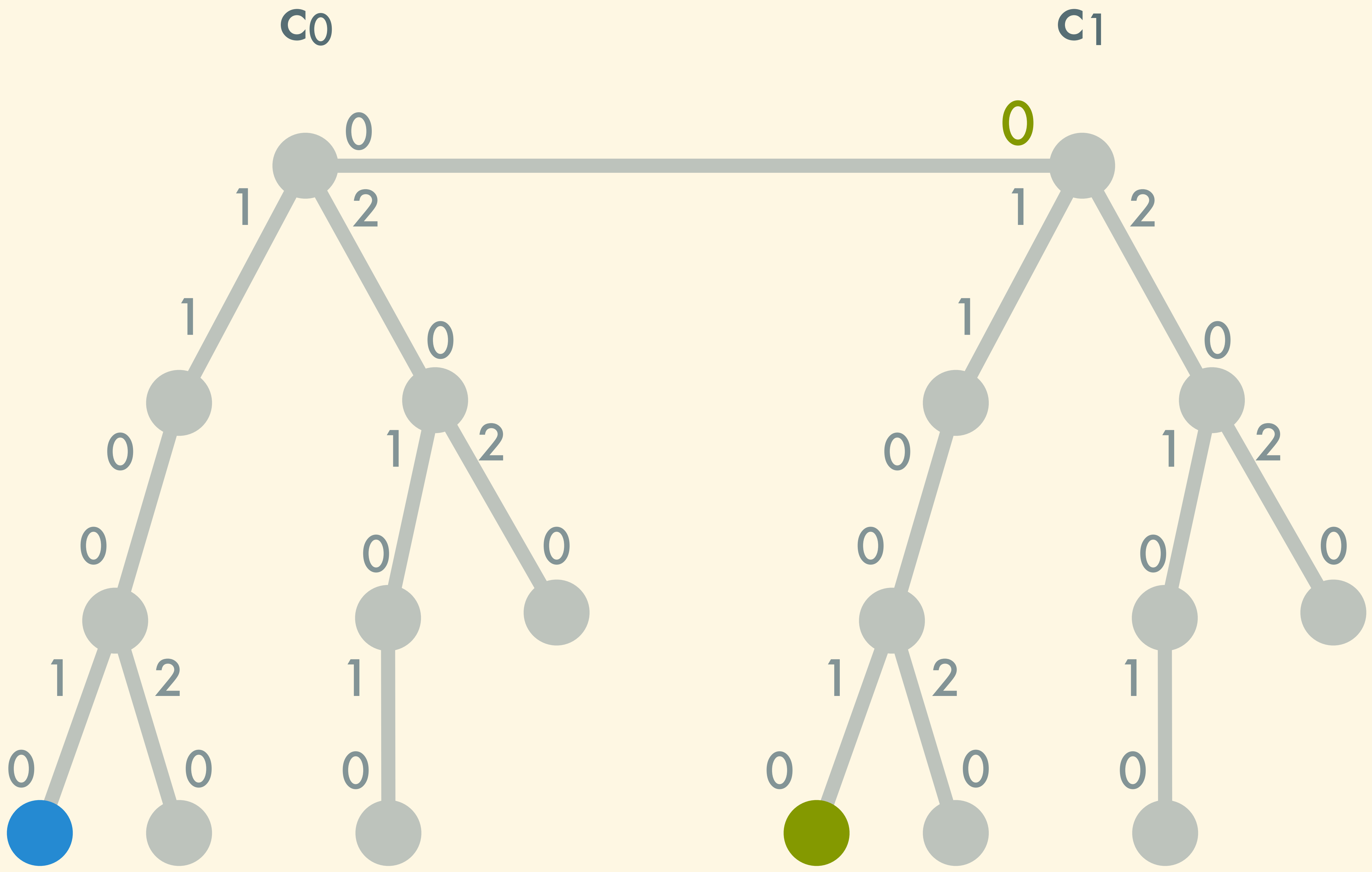
What we need to prove

1. It exists a path that appears in T_0 but not in T_1
2. Nodes in T_i are able to find c_i



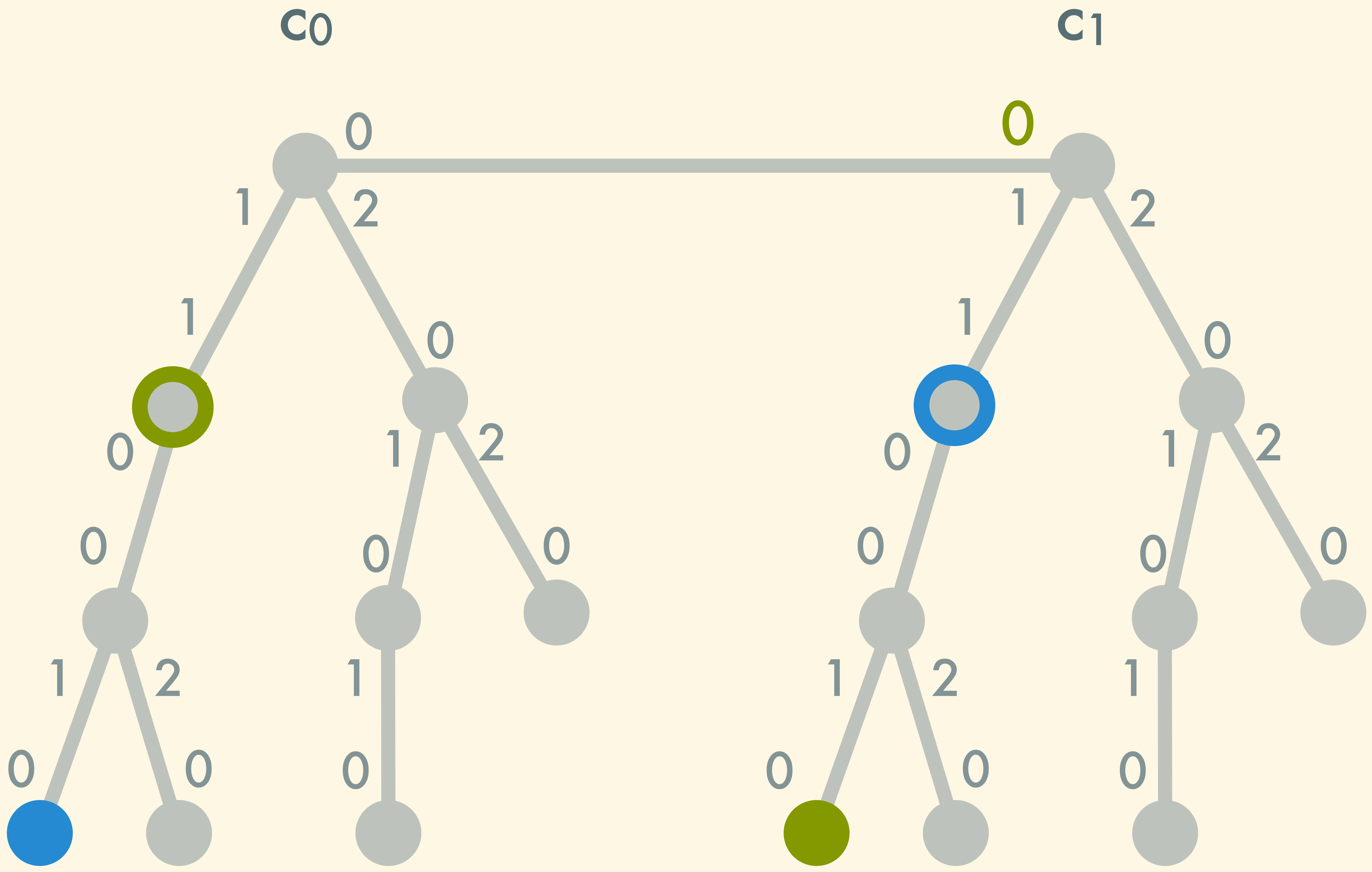
Such a distinctive path exists

- If there is no distinctive path, then
- $T_0 = T_1$
- some nodes have the same view and therefore can't elect correctly
- ie, they will elect symmetrically



Such a distinctive path exists

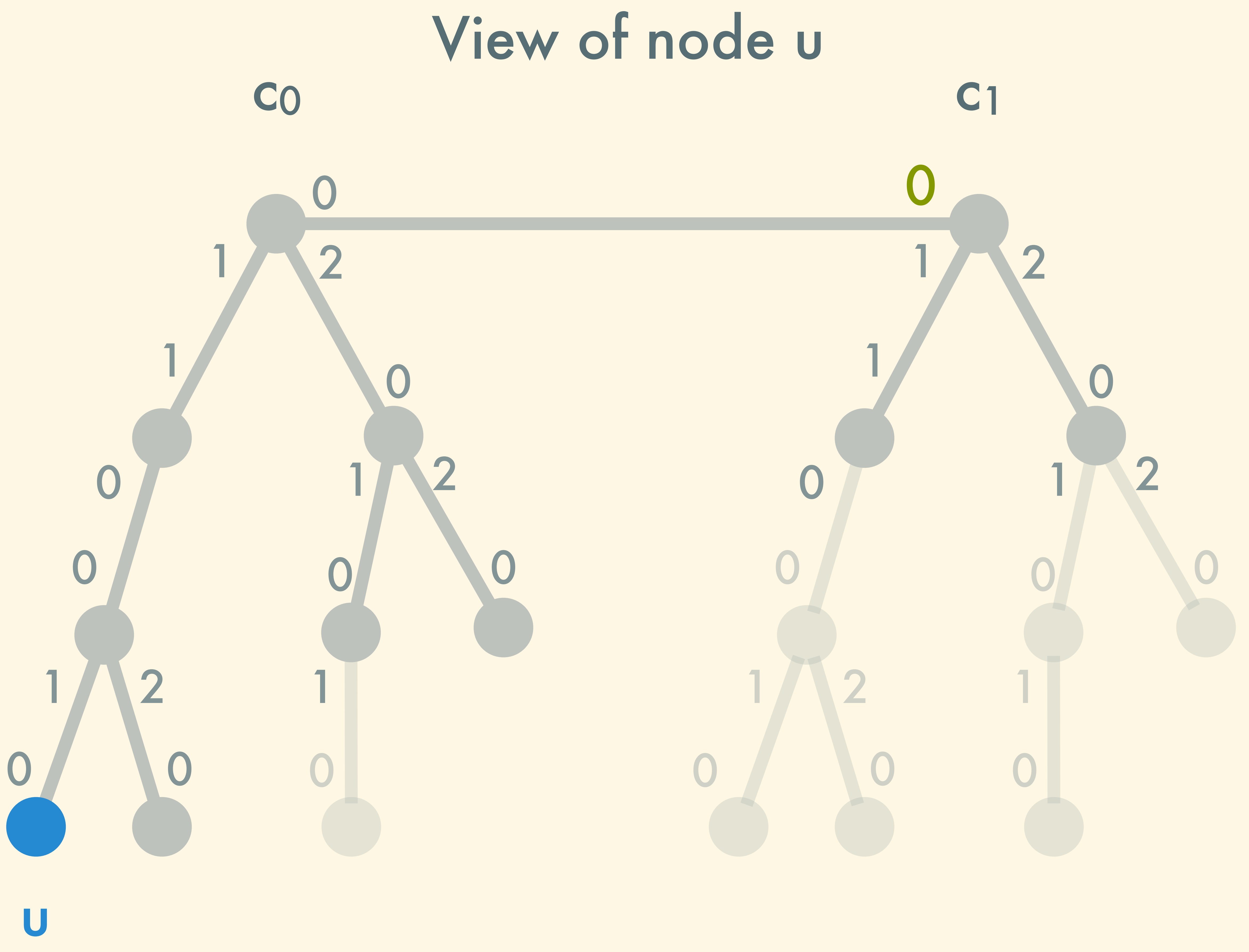
- If there is no distinctive path, then
- $T_0 = T_1$
- some nodes have the same view and therefore can't elect correctly
- ie, they will elect symmetrically



Finding Candidates

How does nodes in T_i find c_i

- Unterminated path: a path that does not end on a leaf
- case1:
  - All unterminated path starting from u go through c_0.
  - c_0 is the furthest node from u through which all unterminated path go.

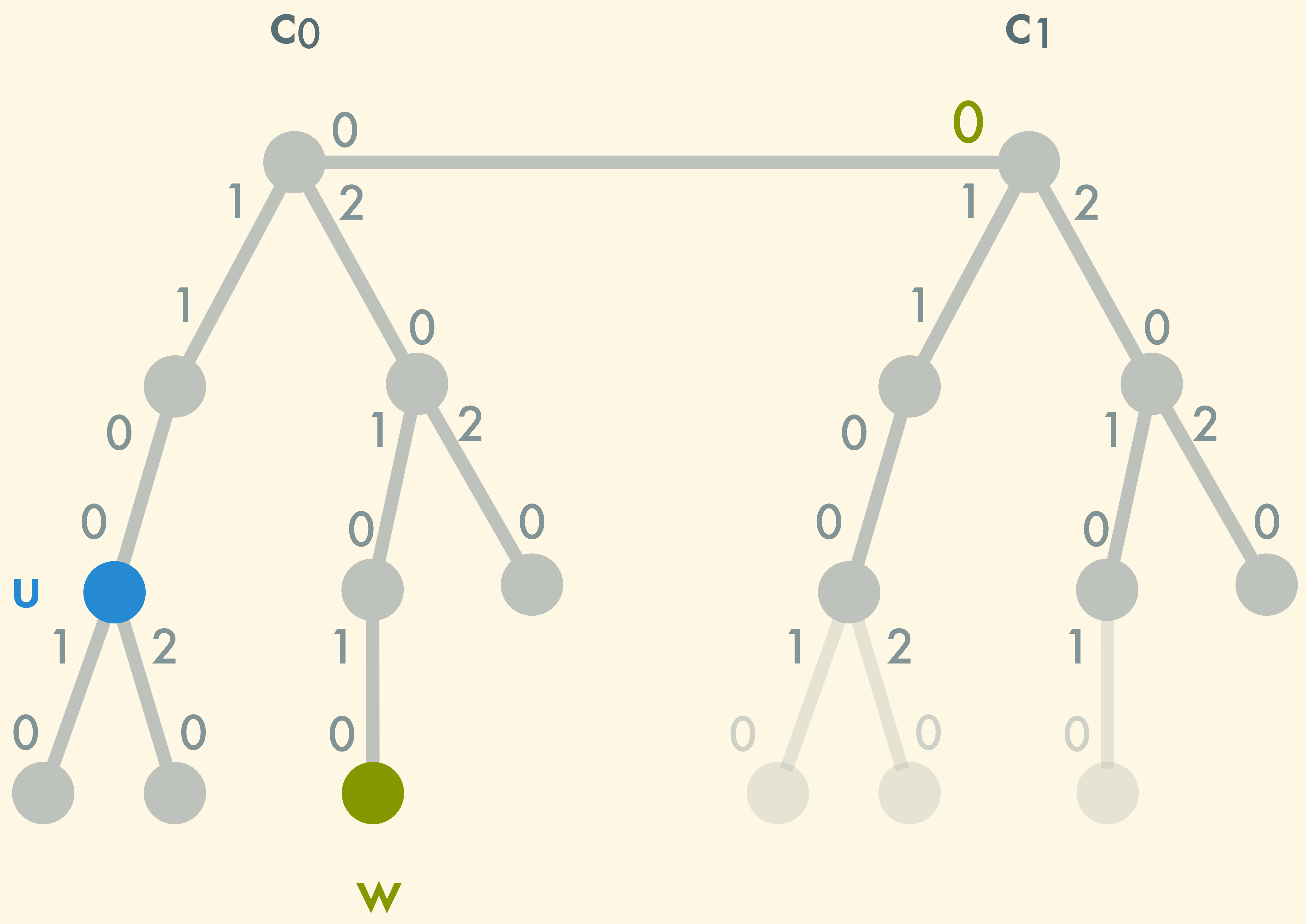


Finding Candidates

How does nodes in T_i find c_i

- Unterminated path: a path that does not end on a leaf
- case2:
  - All unterminated path starting from u go through c_1.
  - In this case it exists w in the view of u such that $d(w,c_1) > d(u,c_1)$
  - The candidate of u is the node one step closer (ie, c_0)

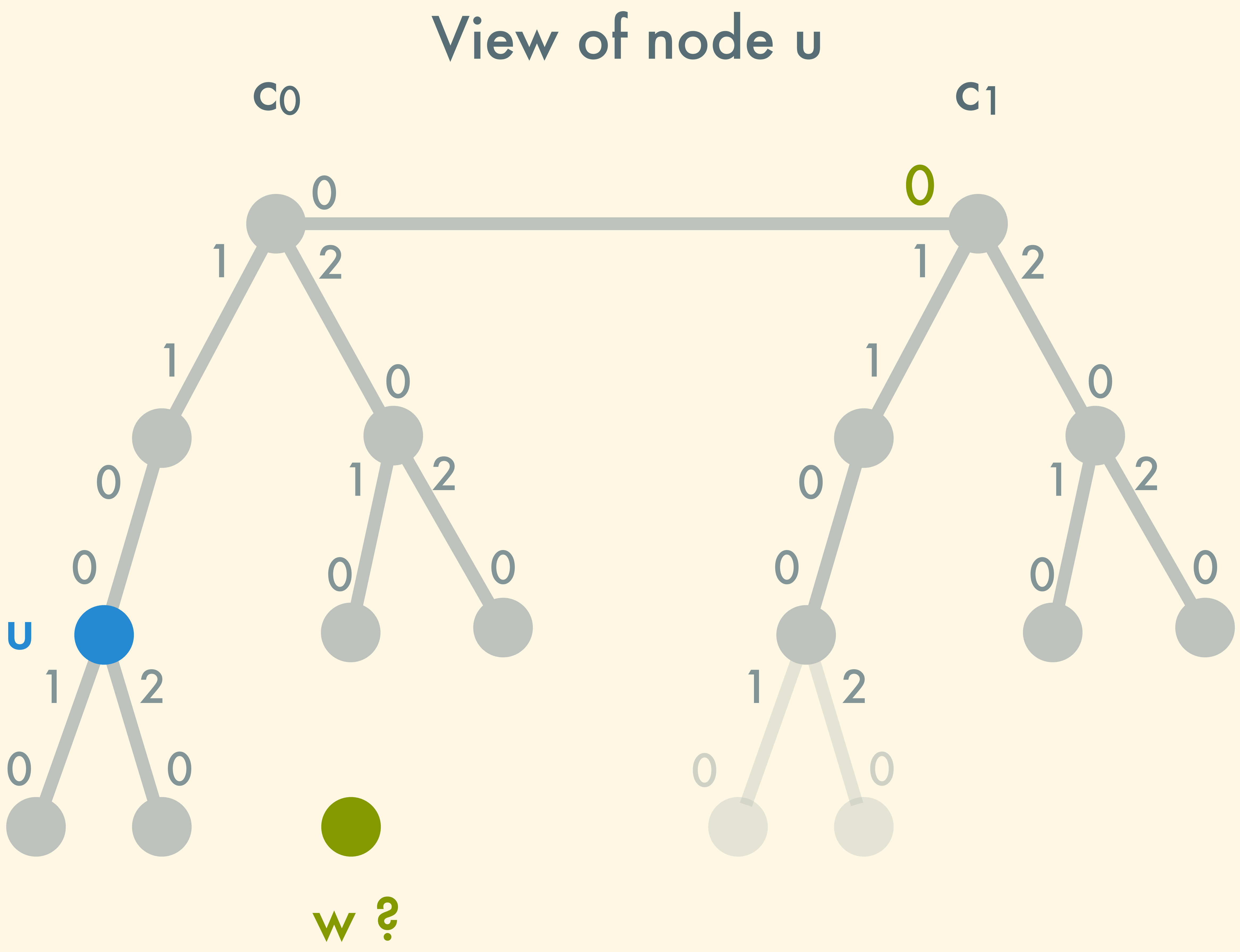
View of node u



Finding Candidates

How does nodes in T_i find c_i

- Unterminated path: a path that does not end on a leaf
- There are more cases that all can be detected using this kind of information.
- Some other cases need an advice to distinguish specific cases (ie, a bit to tell if all unterminated paths of all nodes go through the central edge).

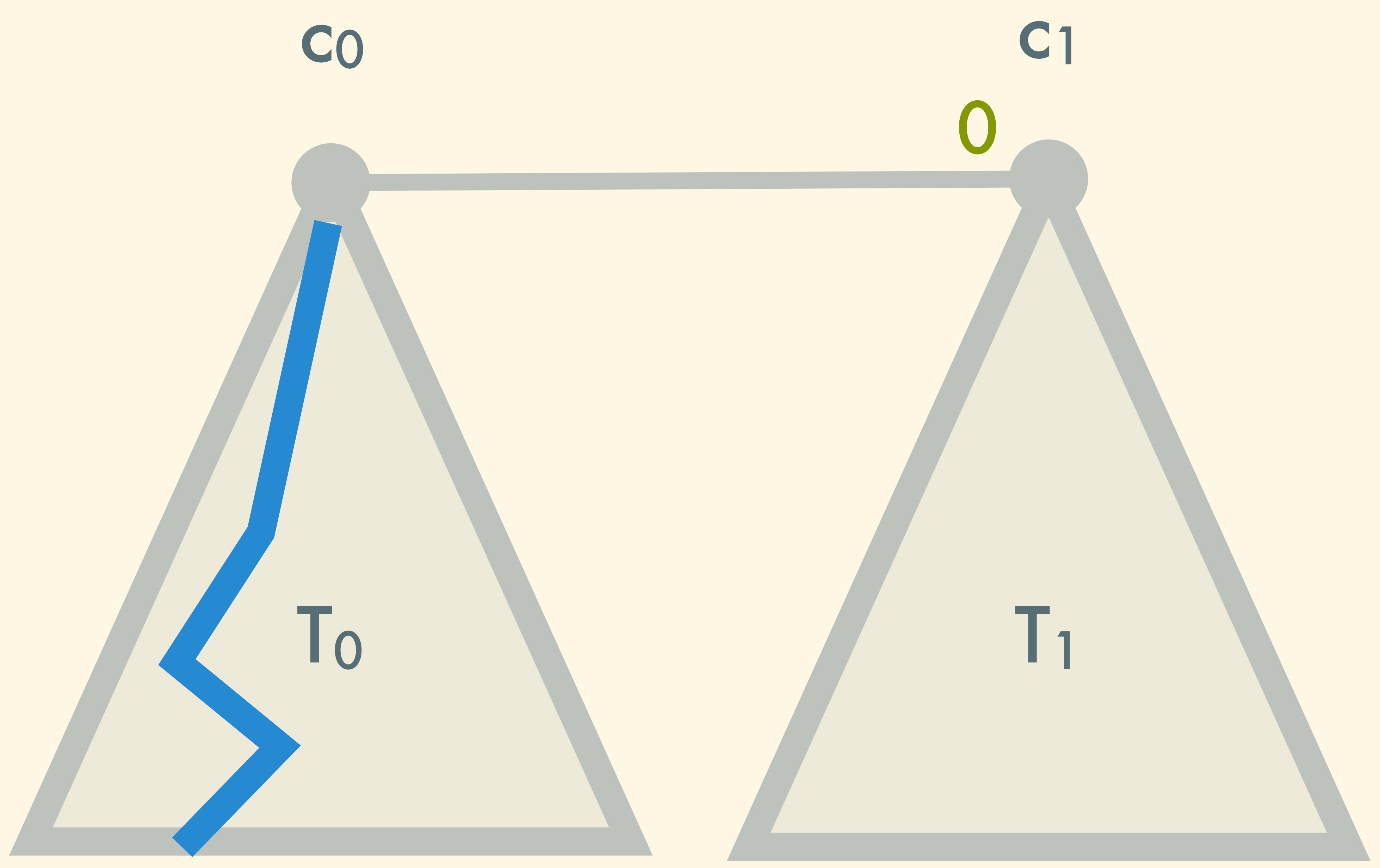


Advice

Advice

- A path that appears in T_0 but not in T_1
- The port number from c_1 to c_0

size: O(D log n)



Reduce advice size

- Build two lexicographically ordered lists L_0 and L_1 of all paths starting from c_0 in T_0 (respc. c_1 in T_1)
- Take as a marker, the first path in L_0 that does not appear in L_1

- Give as an advice:
  - the index of this path: j
  - the first port number that differ in $L_0[j] \neq L_1[j]$: p
  - and its index in $L_0[j]$: k

- This advice size is O(log n)

ODD DIAMETER, TIME D-2

Lower Bound

Lower bound

For a time exactly D-2 and D odd

- Reduction to a pair breaking problem:

1. The pair breaking problem requires Ω(log log Z) bits of advice
2. Use an hypothetical algorithm, ELECT, that solves leader election in time D-2 using o(log log Z) bits to solve the pair breaking problem.

Lower bound

For a time exactly D-2 and D odd

- Reduction to a pair breaking problem:

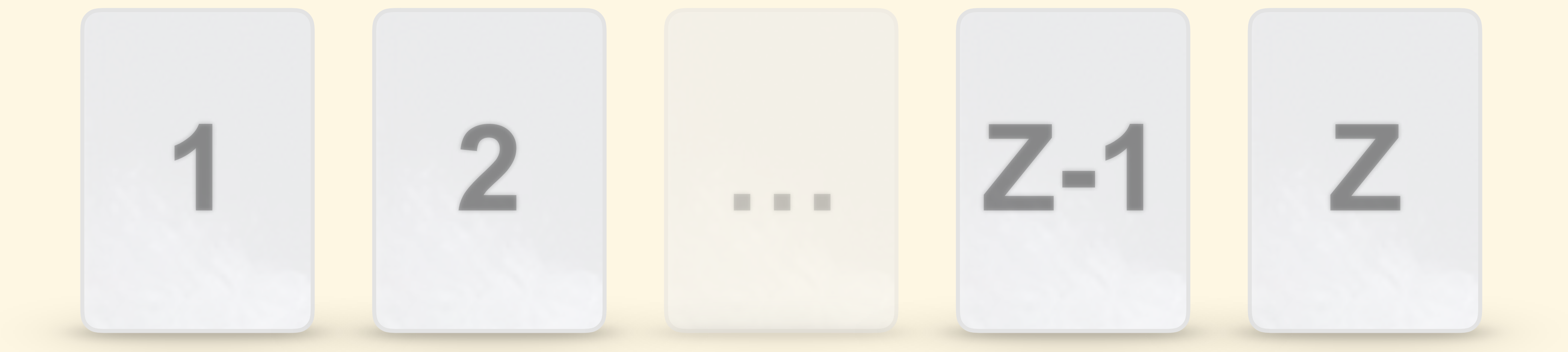
1. The pair breaking problem requires Ω(log log Z) bits of advice

2. Use an hypothetical algorithm, ELECT, that solves leader election in time D-2 using o(log log Z) bits to solve the pair breaking problem.

Pair breaking problem

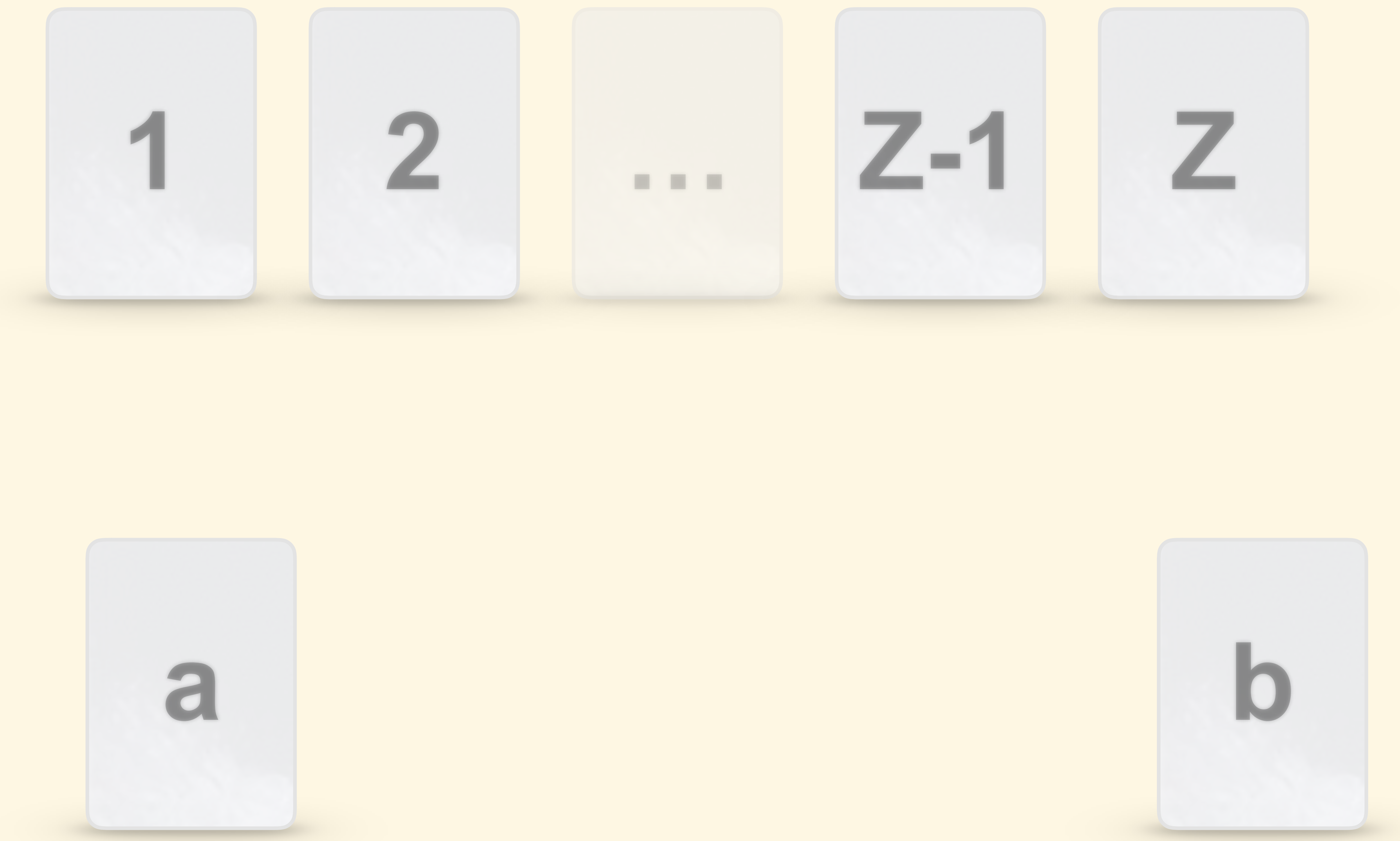
Pair breaking problem

- Integer set S = {1,…, Z}



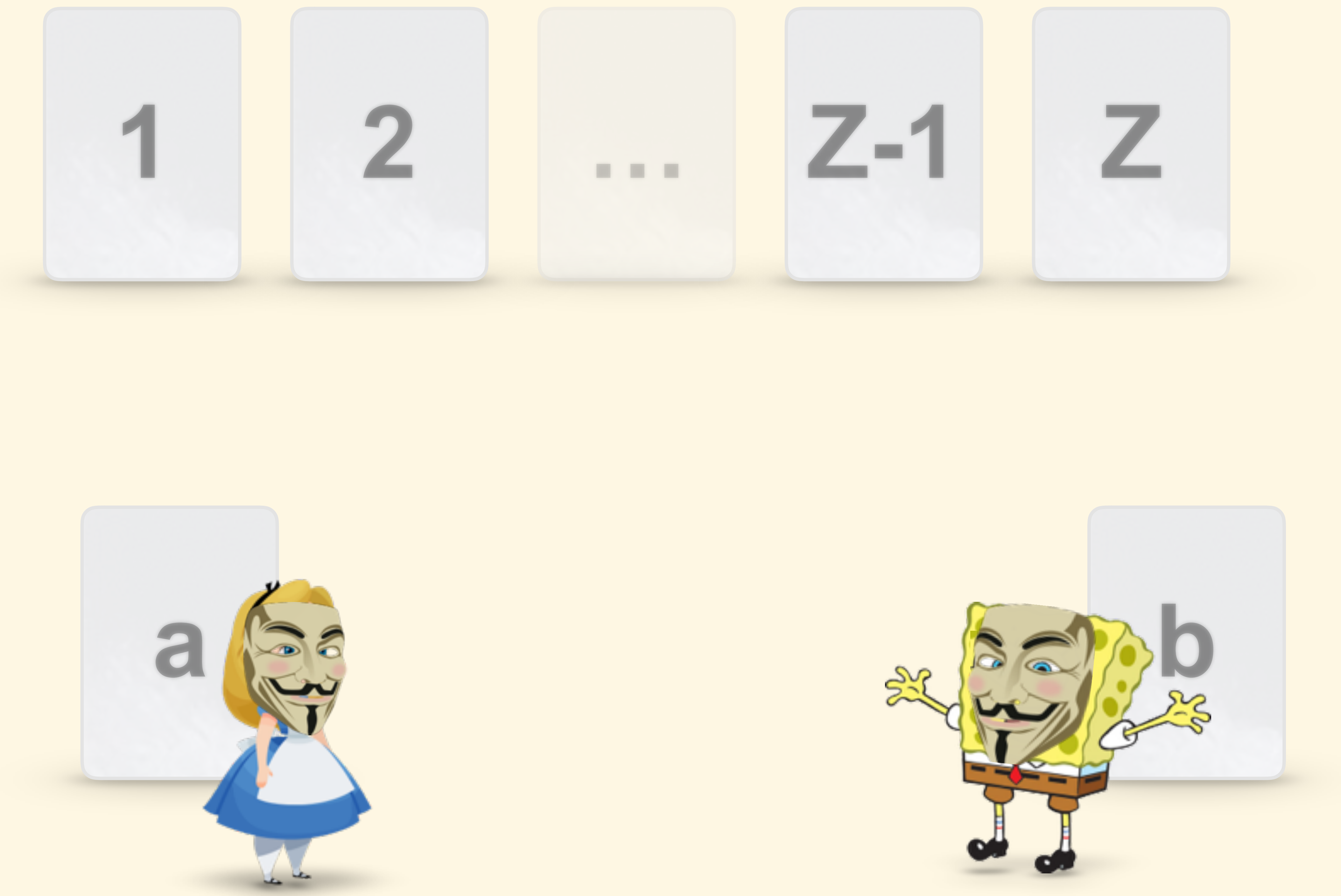
Pair breaking problem

- Integer set S = {1,…, Z}
- Every instance of the problem is a pair (a,b) $\in S^2$, a ≠ b



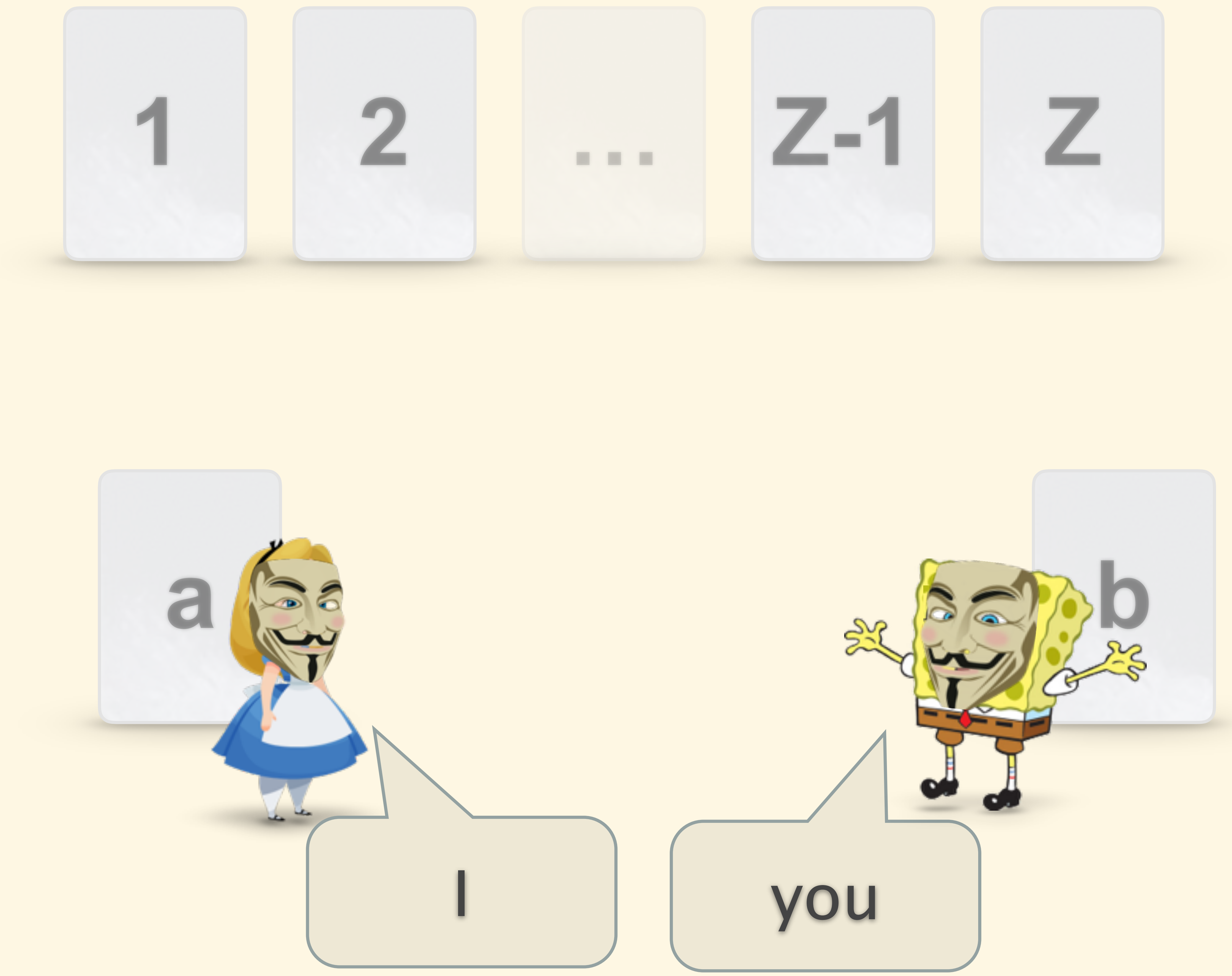
Pair breaking problem

- Integer set S = {1,…, Z}
- Every instance of the problem is a pair (a,b) $\in S^2$, a ≠ b
- Anonymous Players
  - know either a or b
  - must output either "I" or "you"



PAIR BREAKING PROBLEM

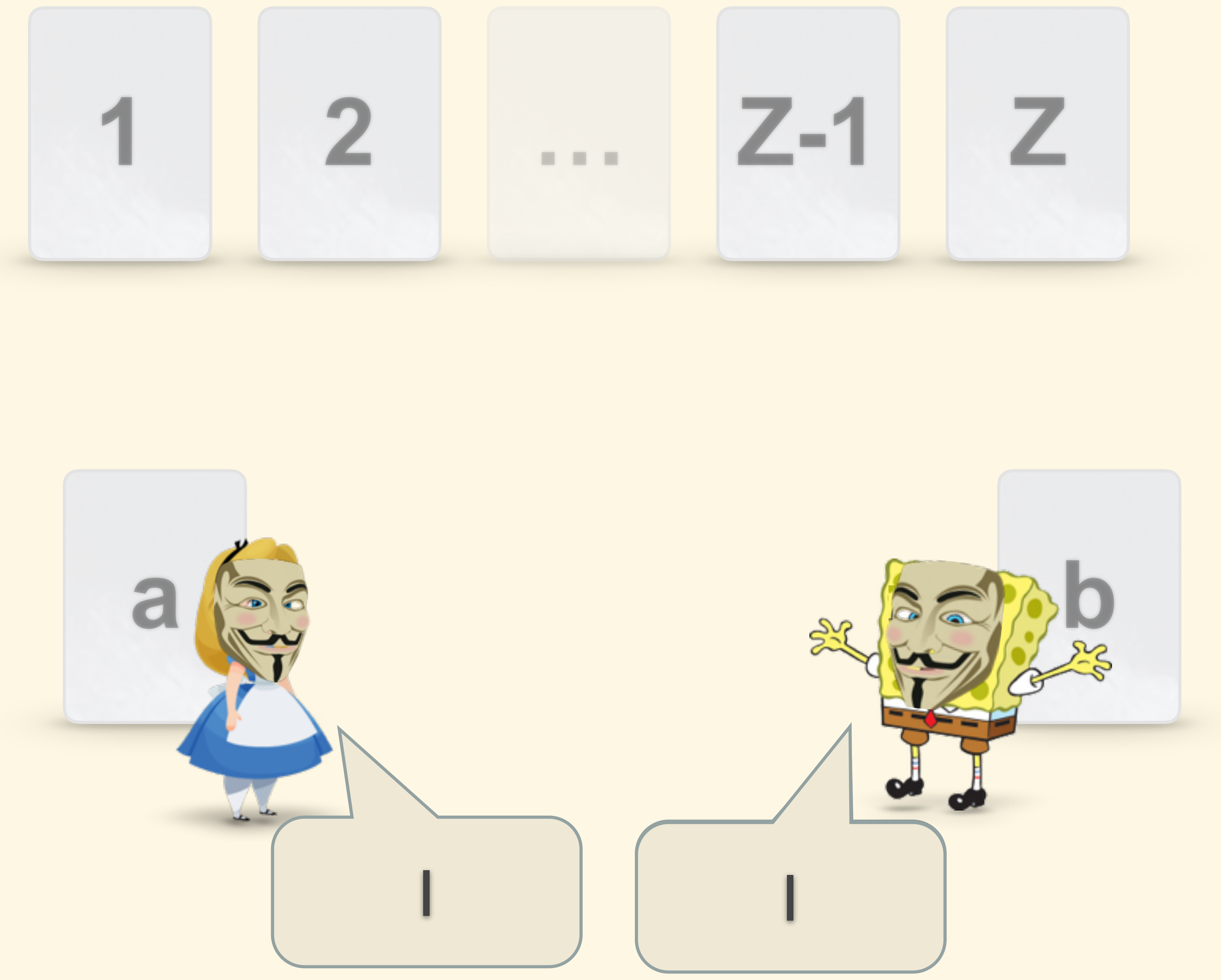
- Integer set $S = \{1, \ldots, Z\}$
- Every instance of the problem is a pair $(a,b) \in S^2$, $a \neq b$
- Anonymous Players
  - know either a or b
  - must output either "I" or "you"



SUCCES

PAIR BREAKING PROBLEM

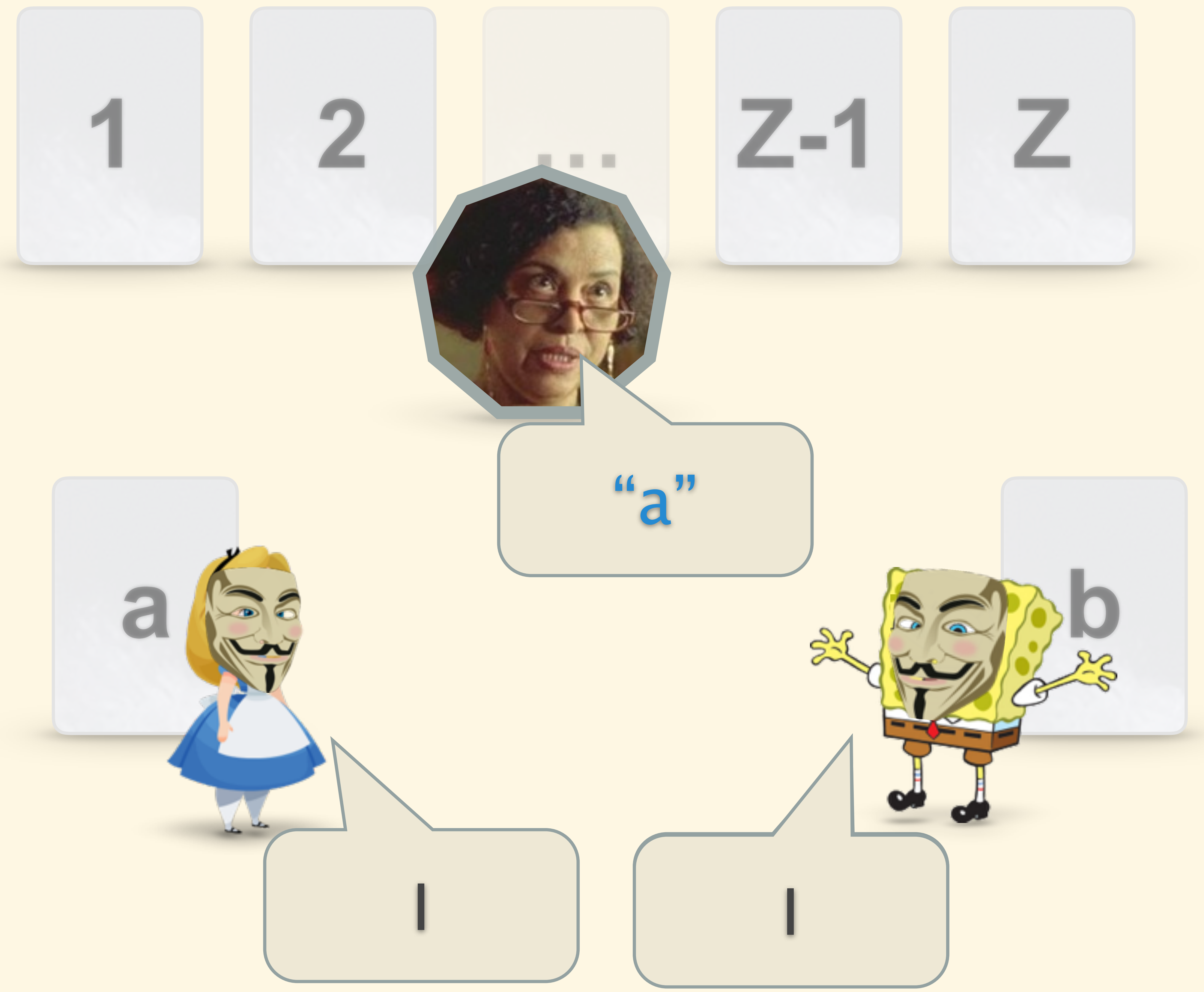
- Integer set S = {1,…, Z}
- Every instance of the problem is a pair (a,b) ∈ S^2, a ≠ b
- Anonymous Players
  - know either a or b
  - must output either "I" or "you"



FAILURE

PAIR BREAKING PROBLEM

- Integer set S = {1,..., Z}
- Every instance of the problem is a pair (a,b) ∈ S^2, a ≠ b
- Anonymous Players
  - know either a or b
  - must output either "I" or "you"
- Oracle
  - knows Z, a and b
  - shout an advice



FAILURE

Pair breaking problem

Question:

What is the minimum size of advice so players can succeed for every pair $(a,b) \in S^2$, $a \neq b$?

First observation:

Giving the position of the first bit that differs in a and b is enough to guaranty success.

Pair breaking problem

a: 0 0 1 0 0 0 1 1 1 0 0 1 0 (you)

b: 0 0 1 0 1 0 0 1 0 1 1 1 1 (I)

For an integer set of size Z,
the advice size is O(log log Z)

Is Ω(log log Z) the lower bound on this game?

Pair breaking problem

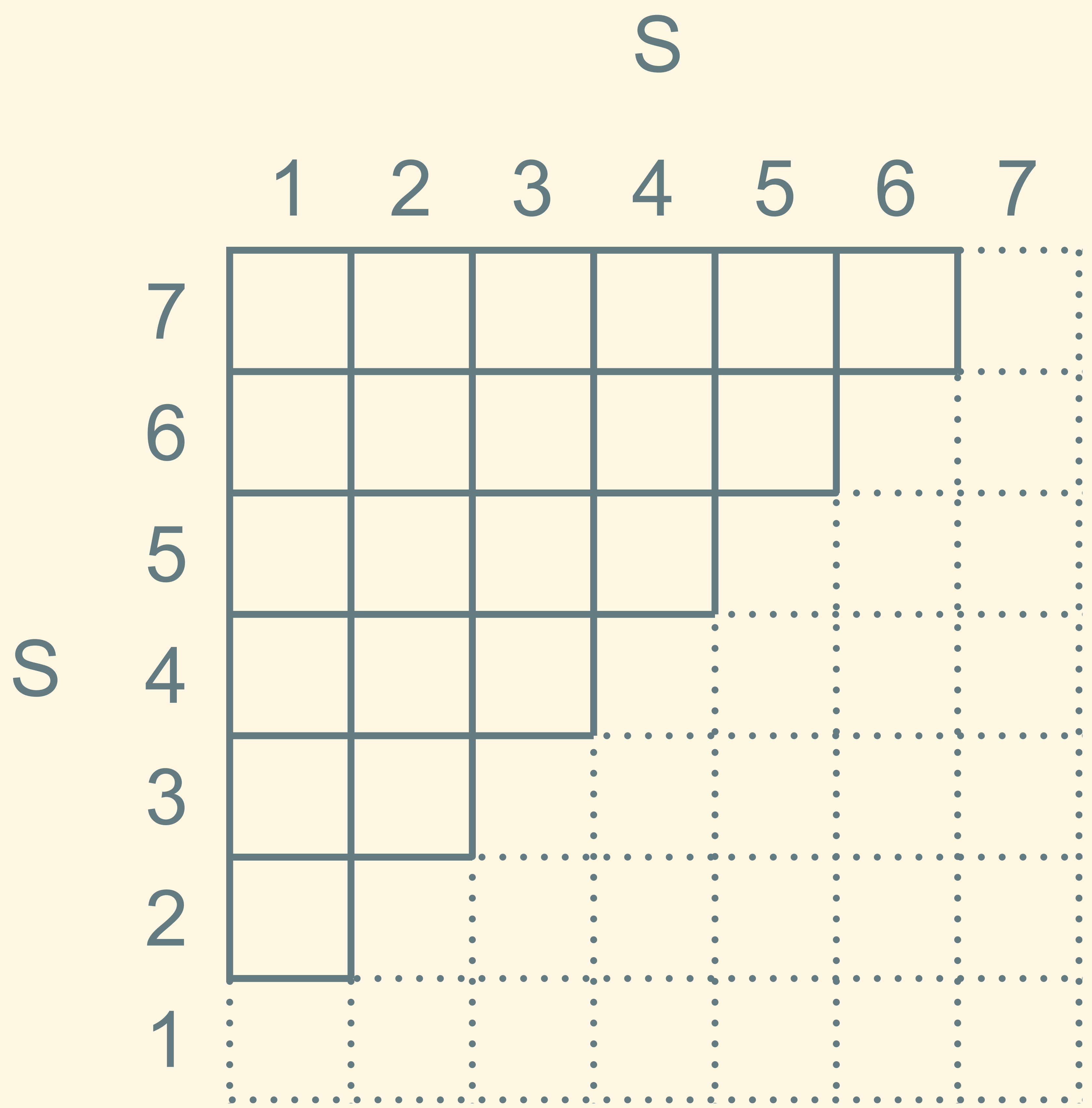
Coloration of a grid

- Integer set $S = \{1,\ldots,Z\}$

Pair breaking problem

Coloration of a grid

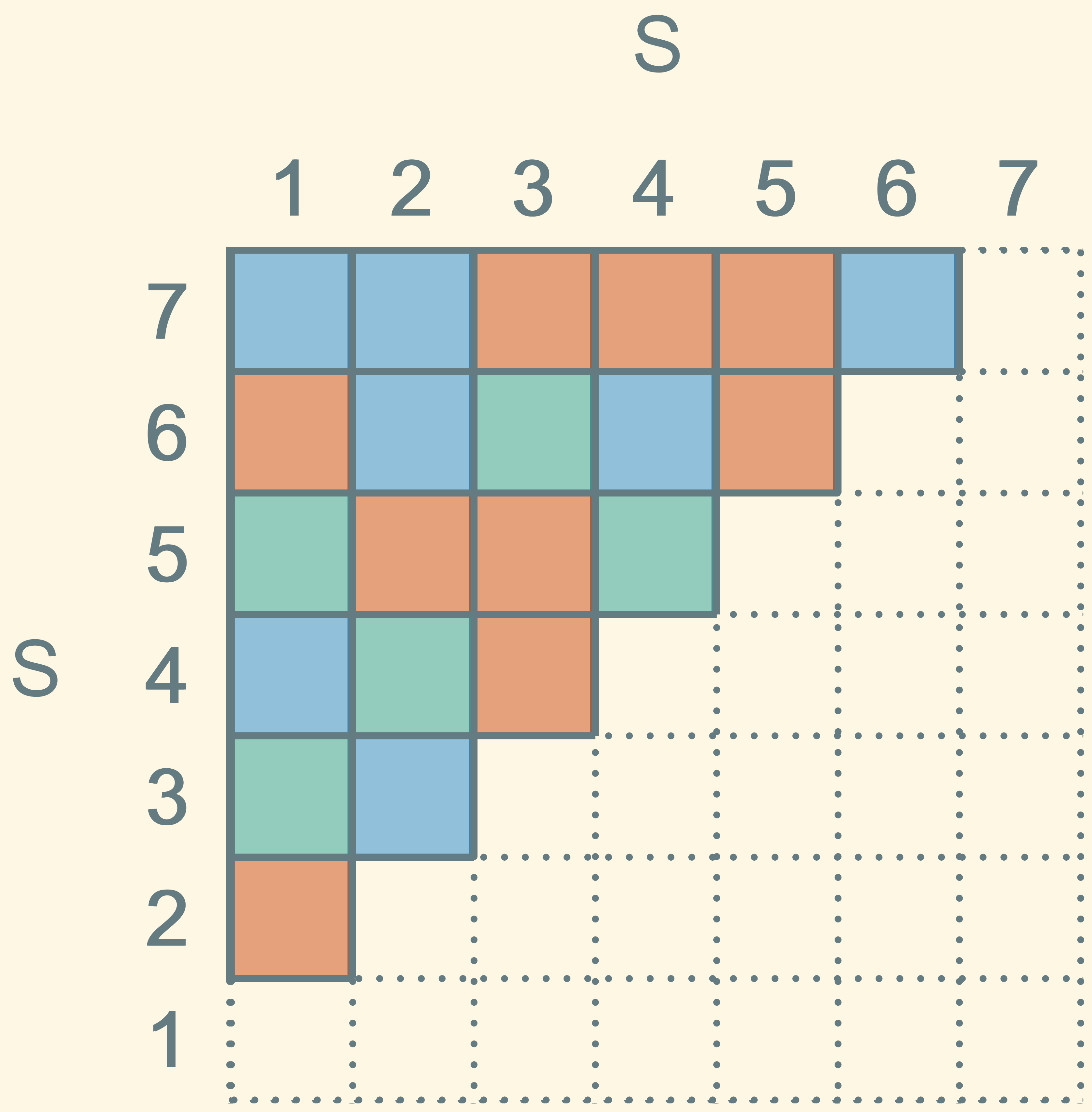
- Integer set $S = \{1,\ldots,Z\}$
- Every instance of the problem is a a cell (a pair $(a,b) \in S^2$, $a < b$)



PAIR BREAKING PROBLEM

Coloration of a grid

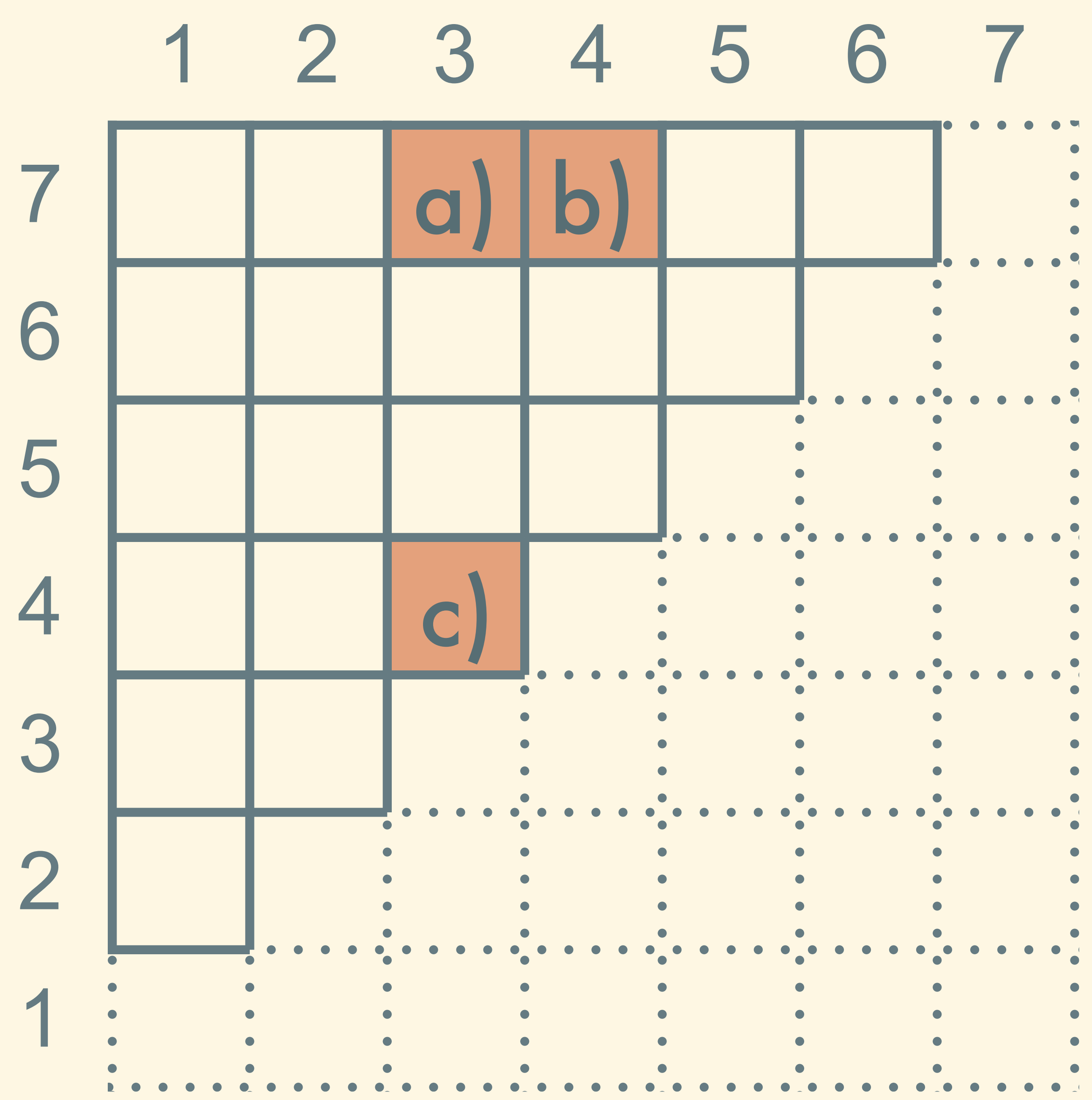
- Integer set $S = \{1,\ldots,Z\}$
- Every instance of the problem is a a cell (*a pair* $(a,b) \in S^2$, $a < b$)
- Oracle
  - knows Z, a and b
  - shout an advice: color every cell



Pair breaking problem

Coloration rules

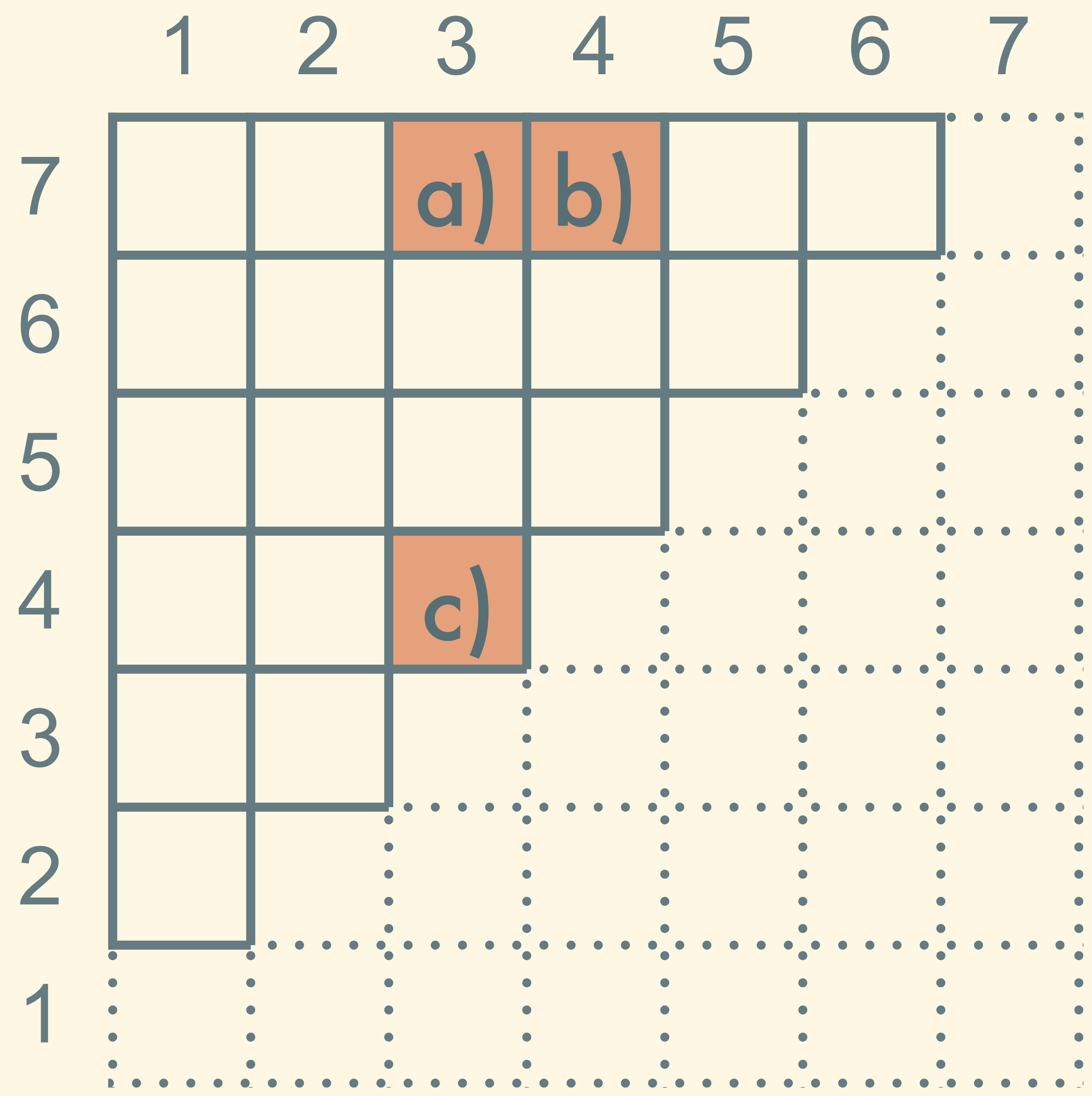
Invalid coloration: no algorithm can solve the problem if this pattern appears.



Pair breaking problem

Coloration rules

Invalid coloration: no algorithm can solve the problem if this pattern appears.

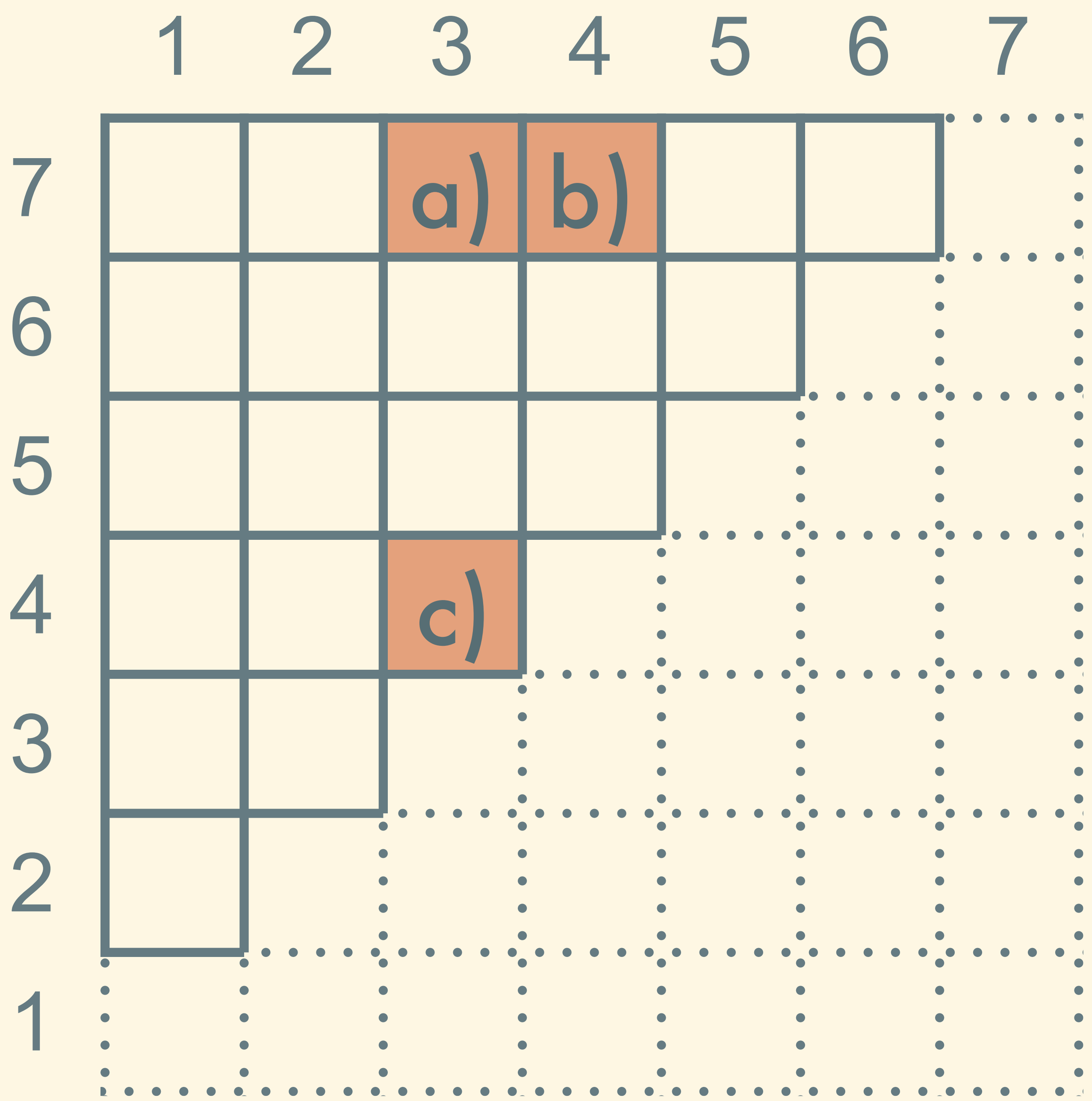


a)

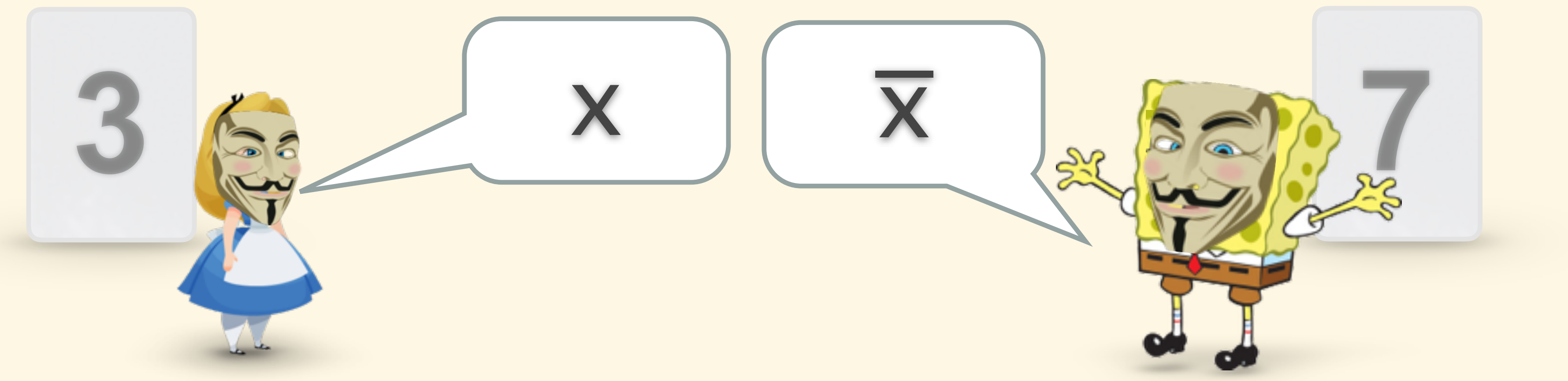
Pair breaking problem

Coloration rules

Invalid coloration: no algorithm can solve the problem if this pattern appears.



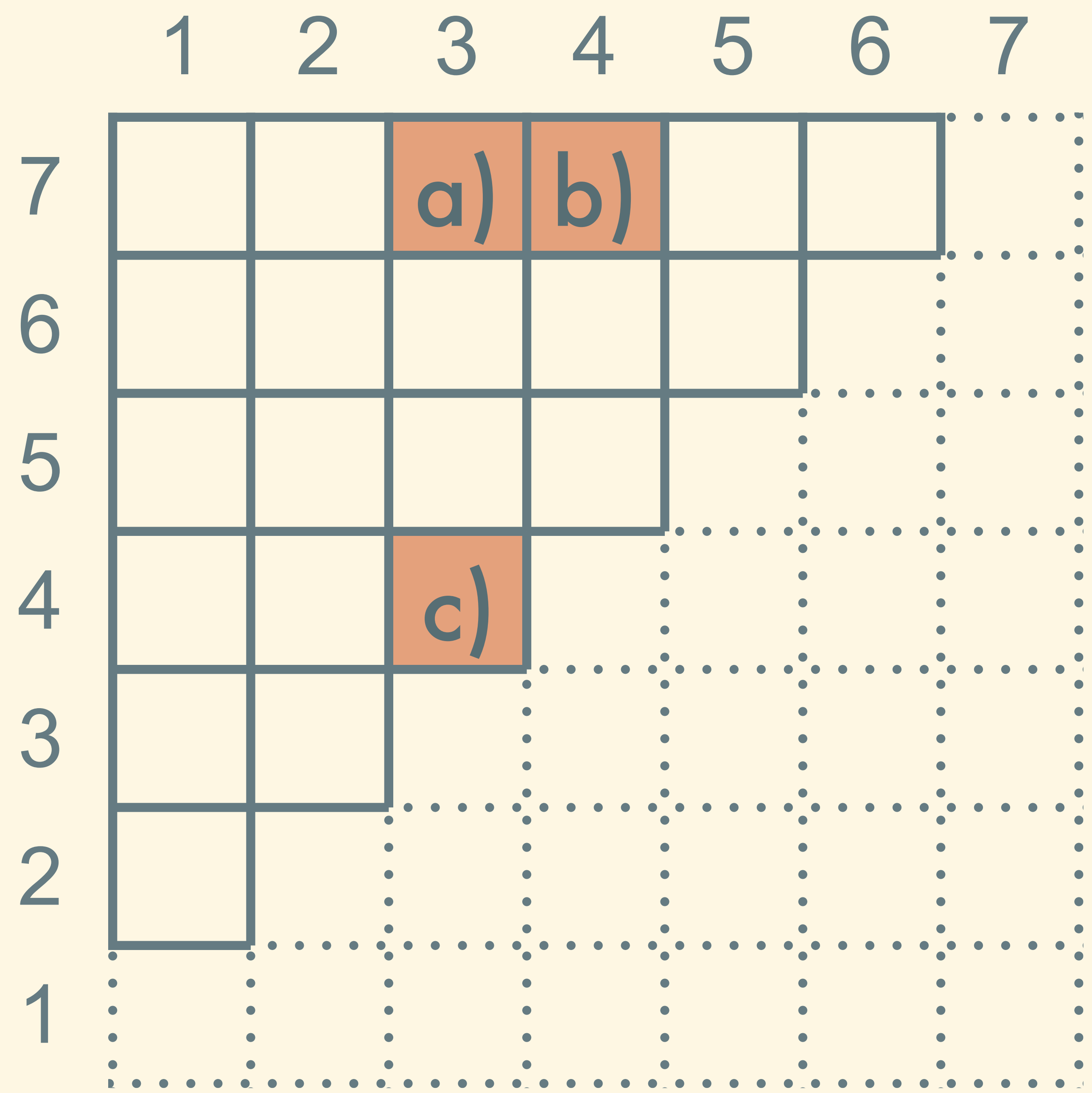
a)

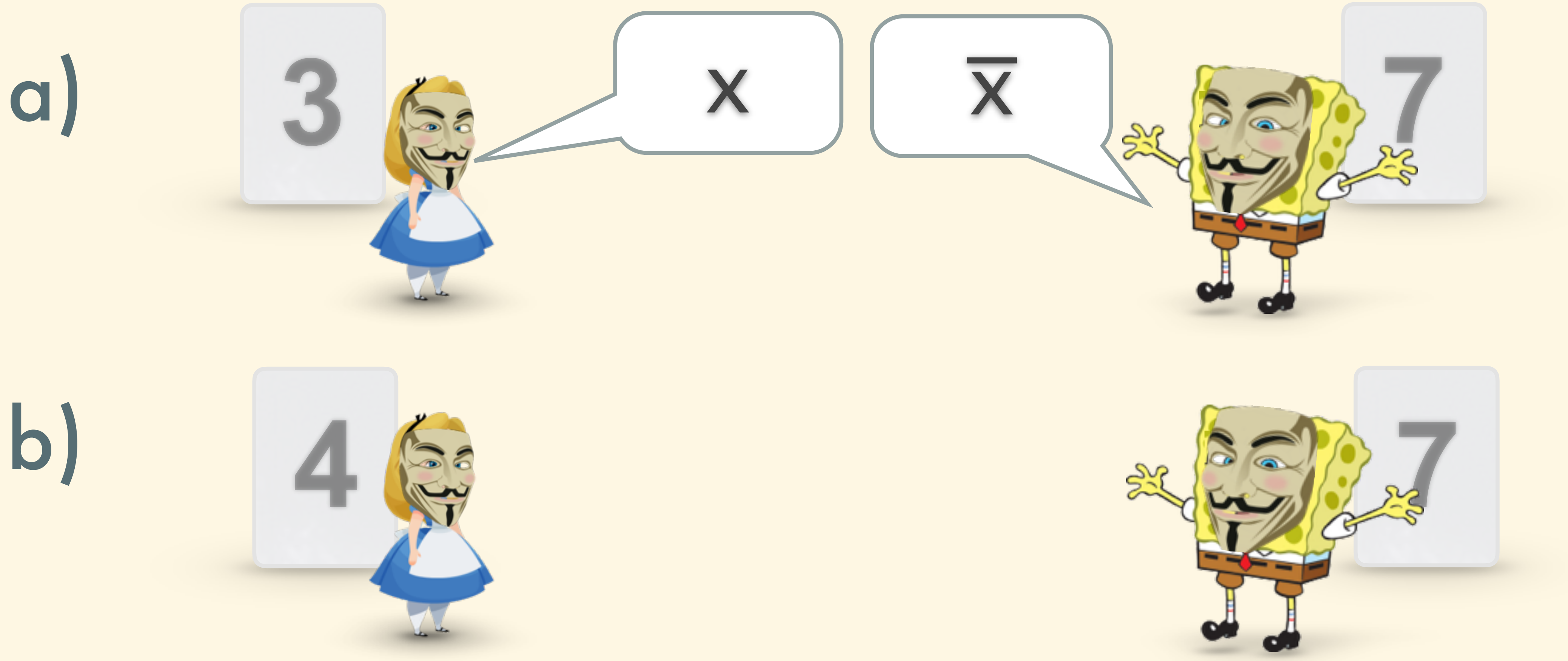


PAIR BREAKING PROBLEM

Coloration rules

Invalid coloration: no algorithm can solve the problem if this pattern appears.

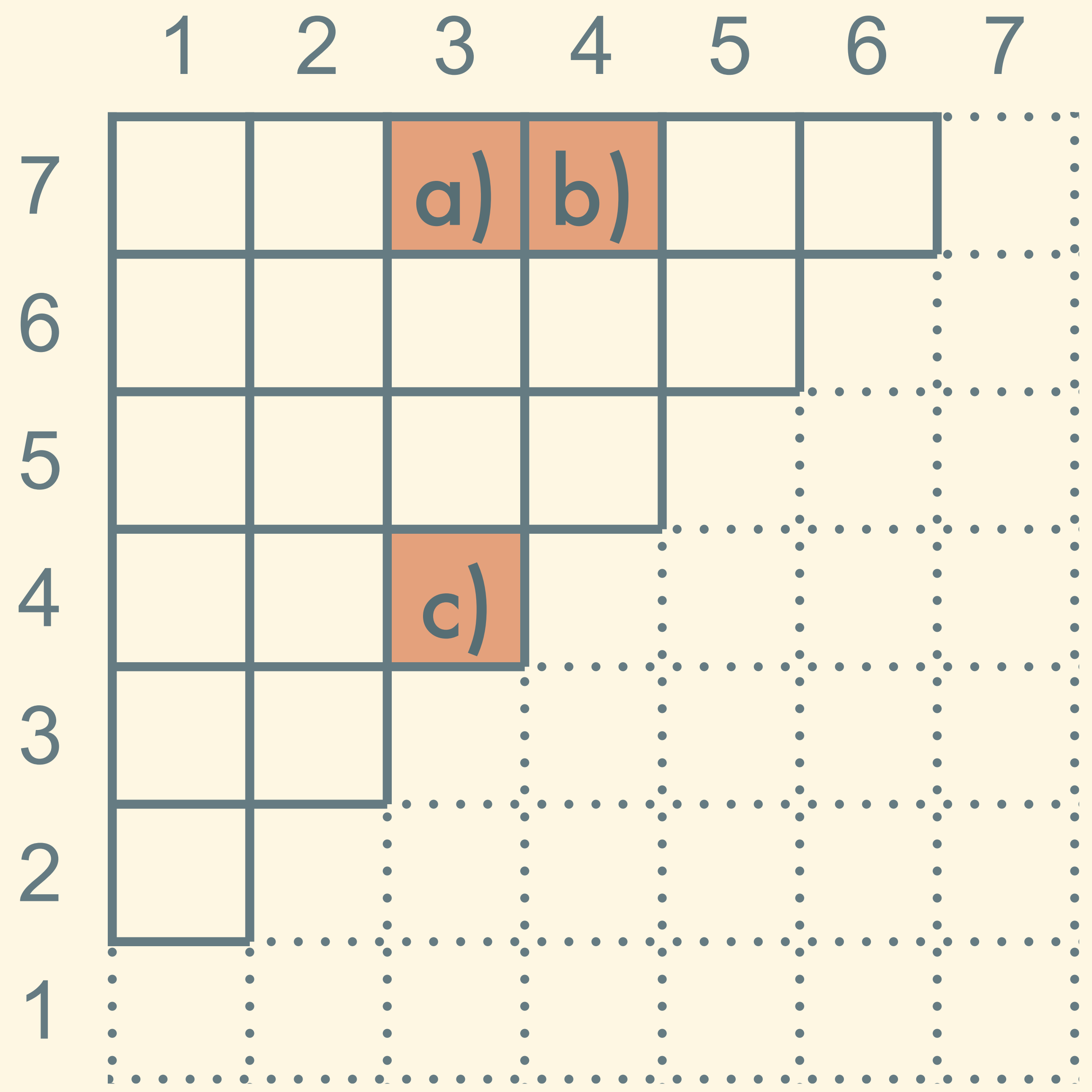


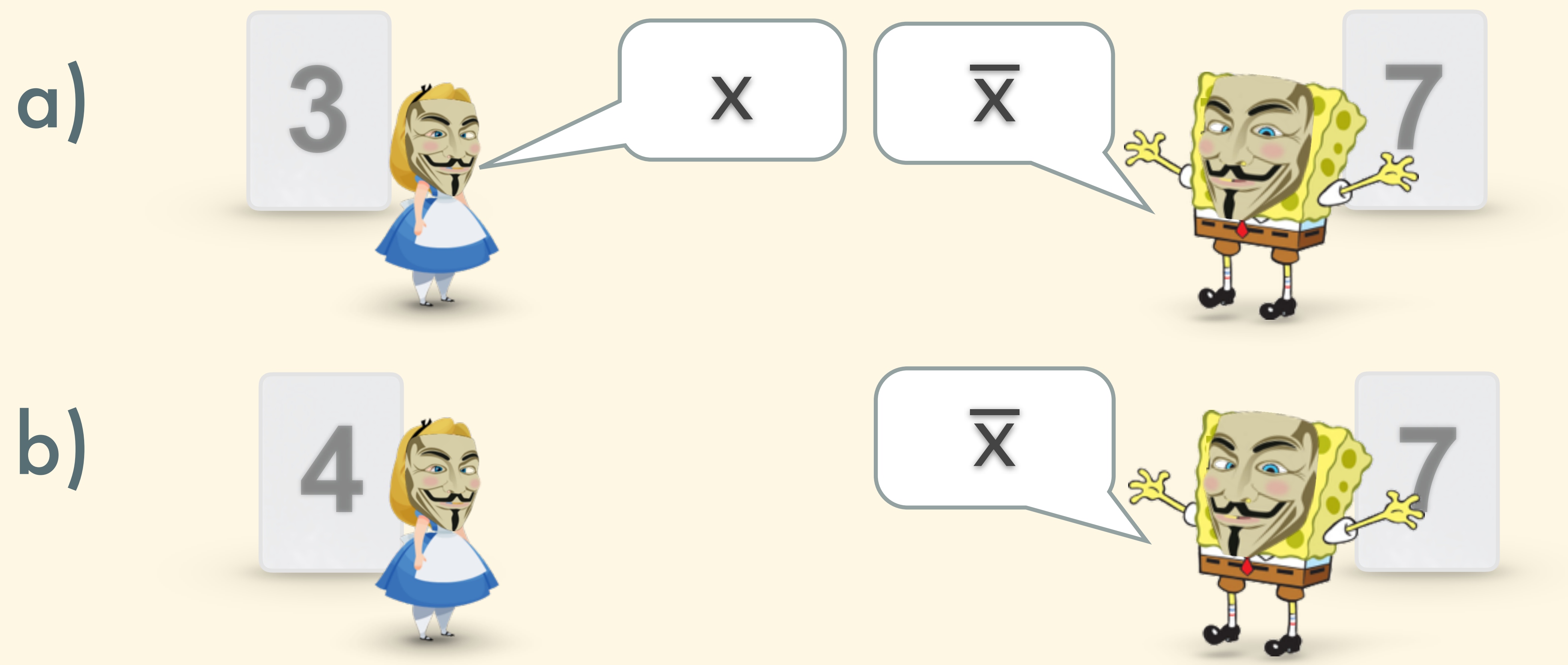


Pair breaking problem

Coloration rules

Invalid coloration: no algorithm can solve the problem if this pattern appears.

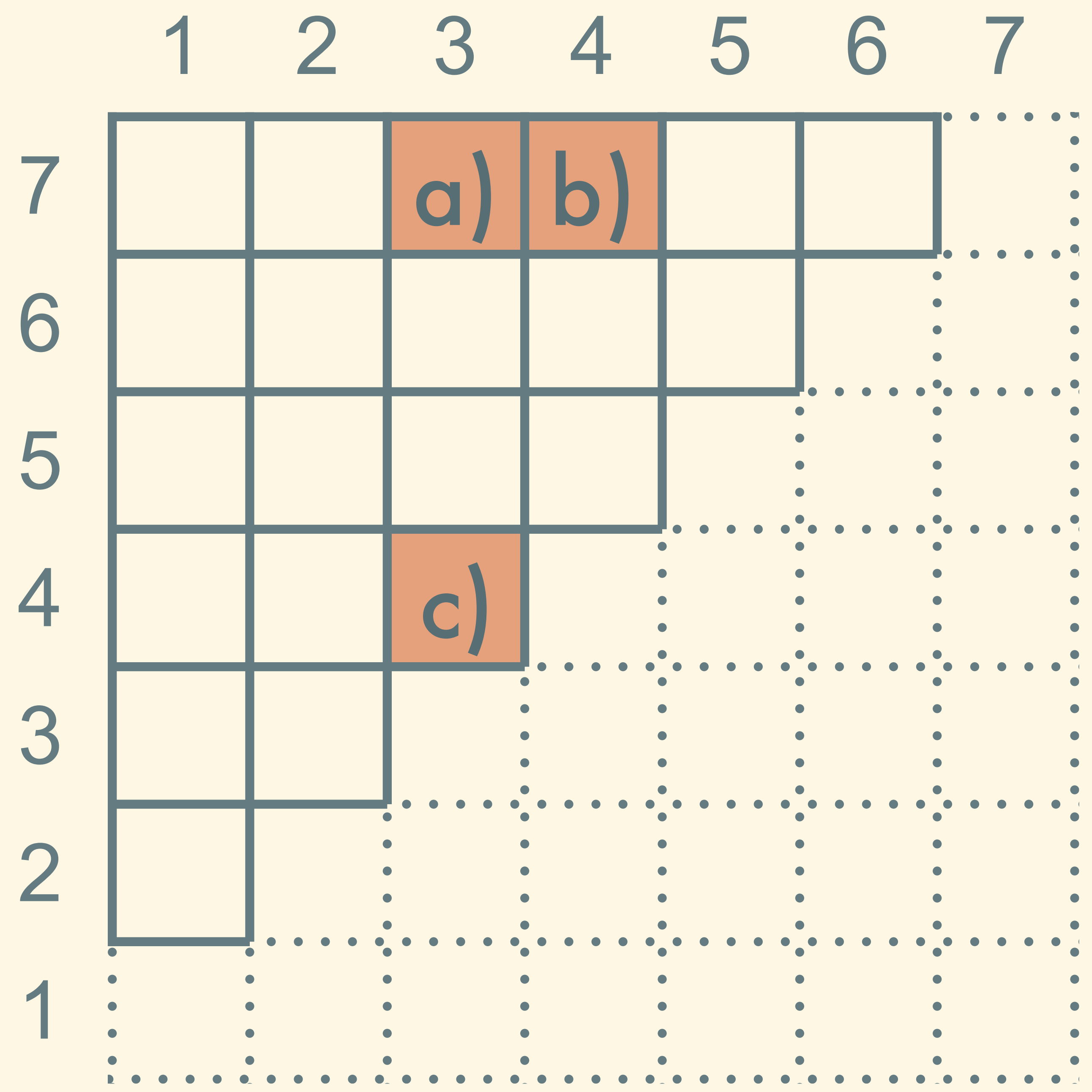


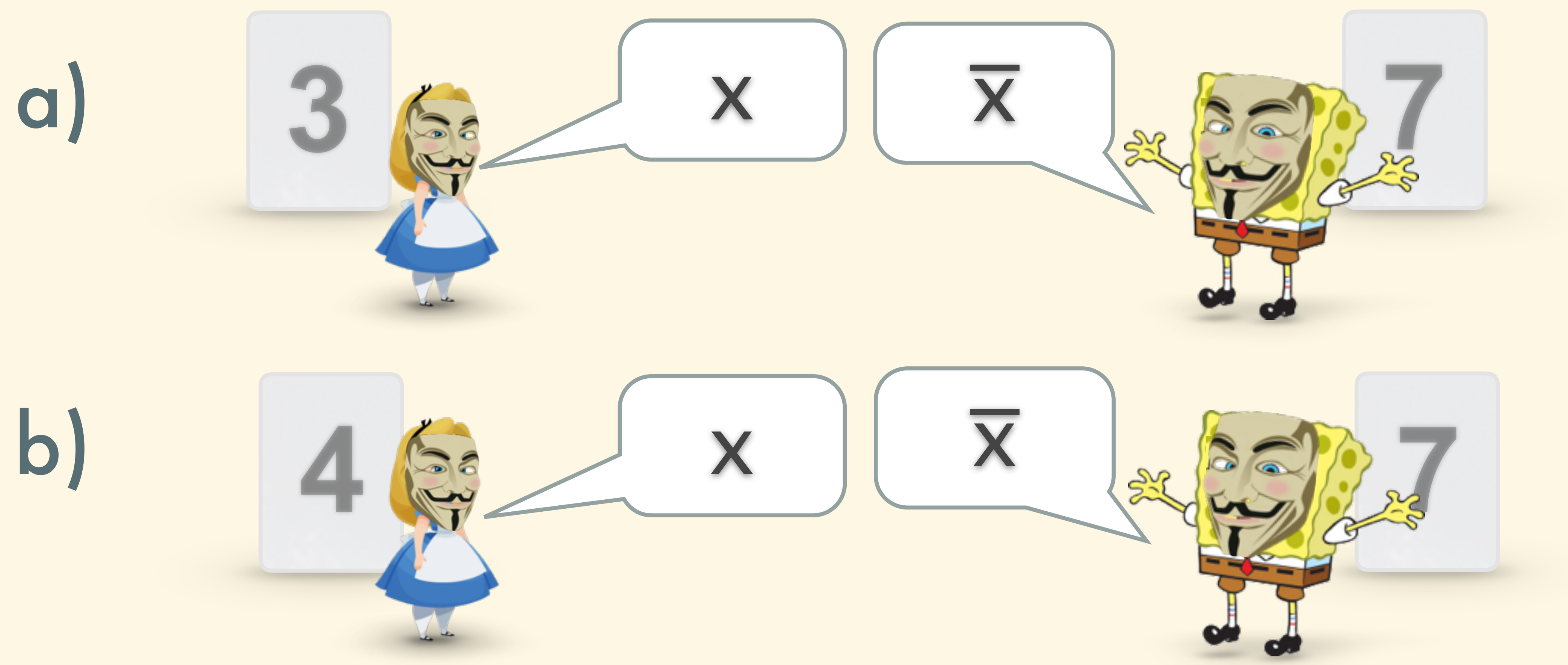


Pair breaking problem

Coloration rules

Invalid coloration: no algorithm can solve the problem if this pattern appears.

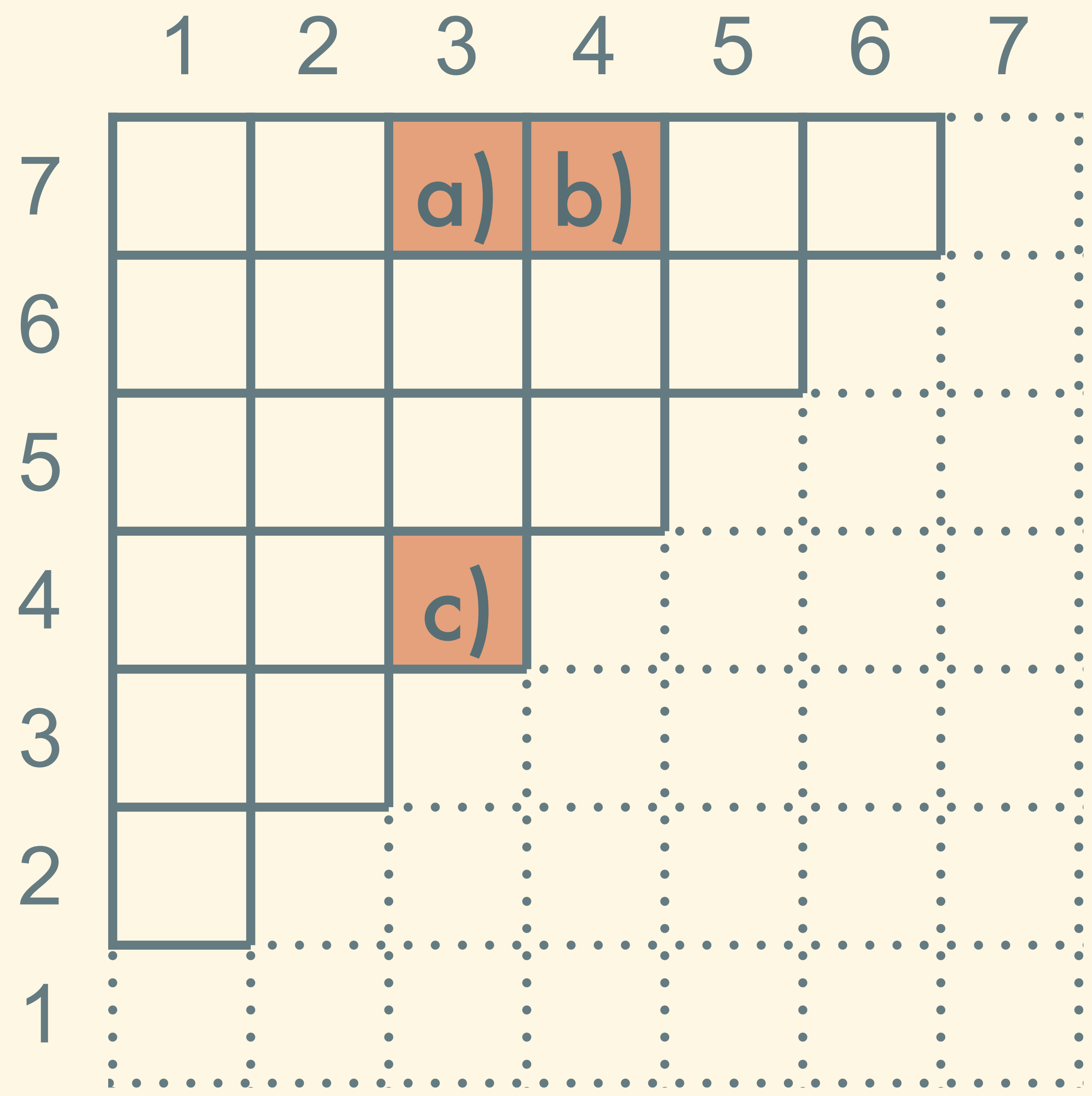


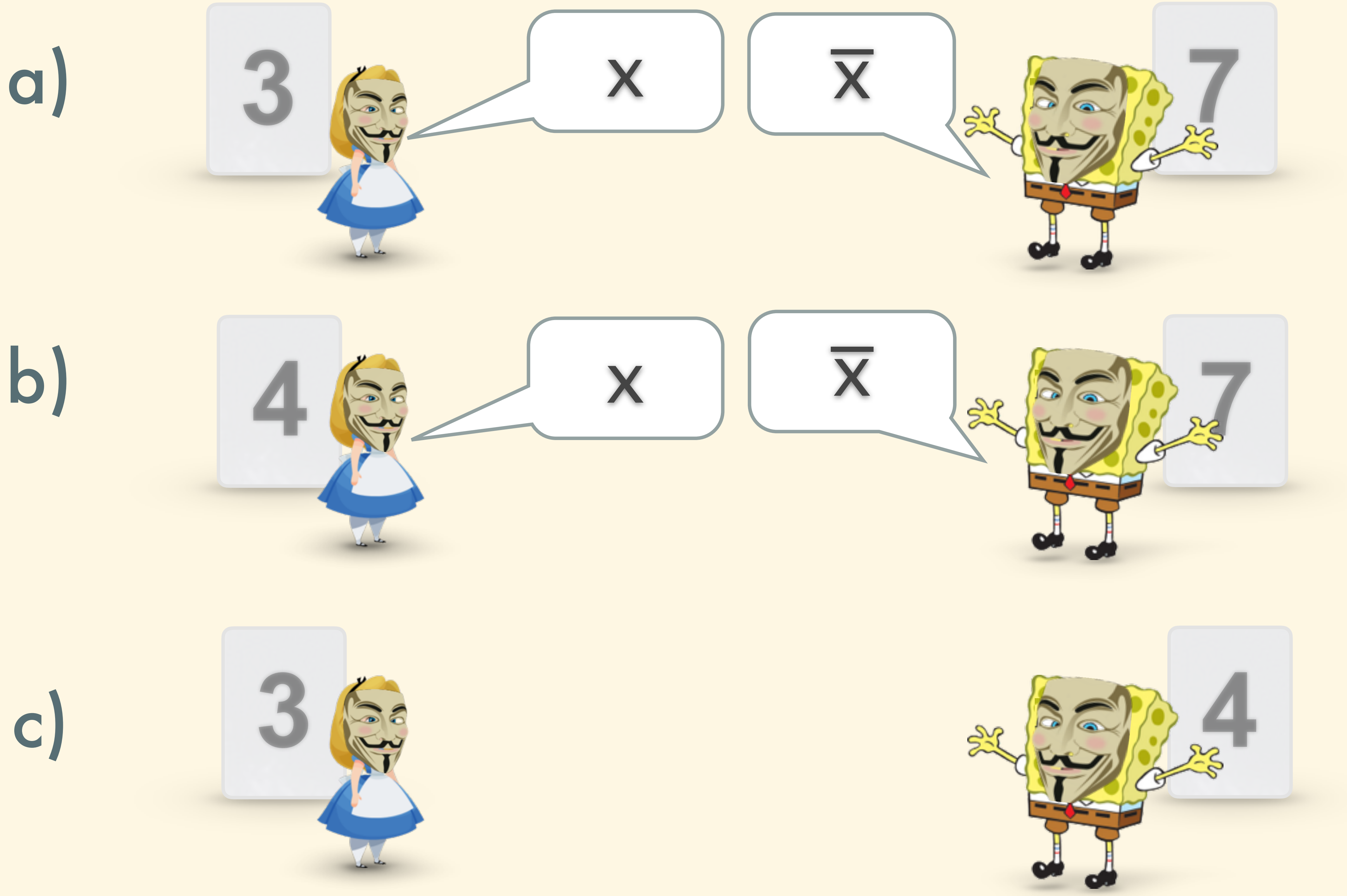


Pair breaking problem

Coloration rules

Invalid coloration: no algorithm can solve the problem if this pattern appears.

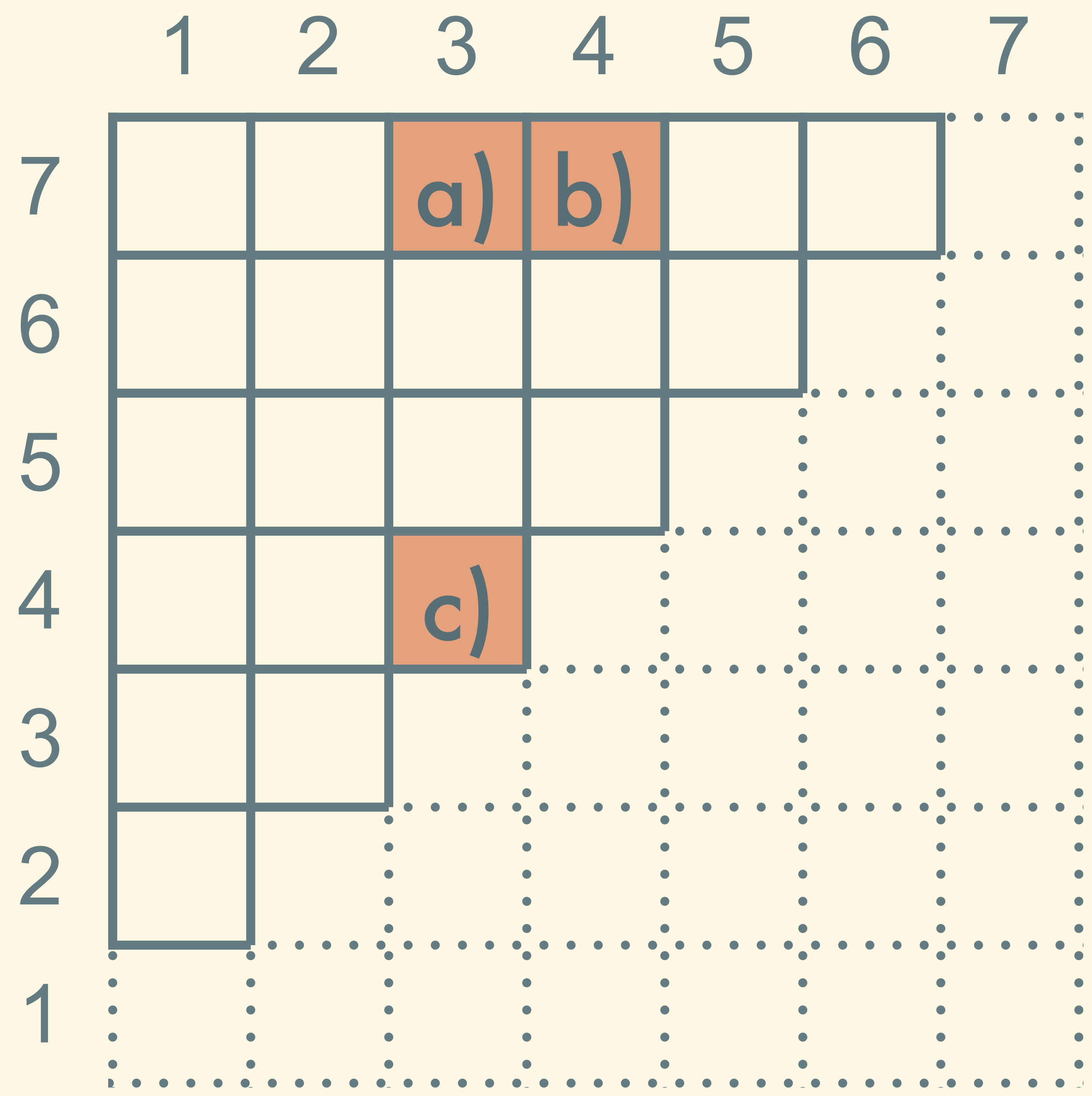


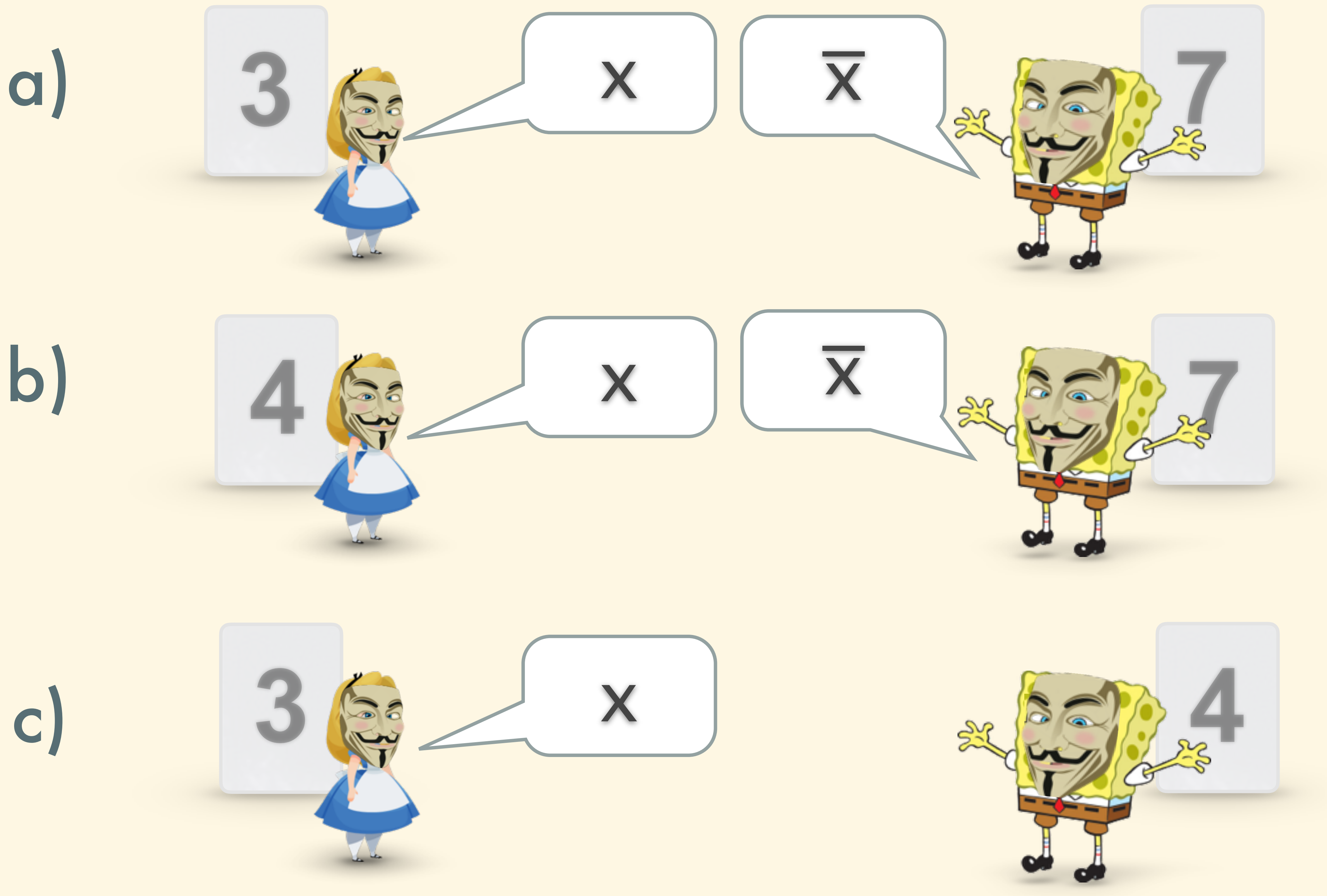


Pair breaking problem

Coloration rules

Invalid coloration: no algorithm can solve the problem if this pattern appears.

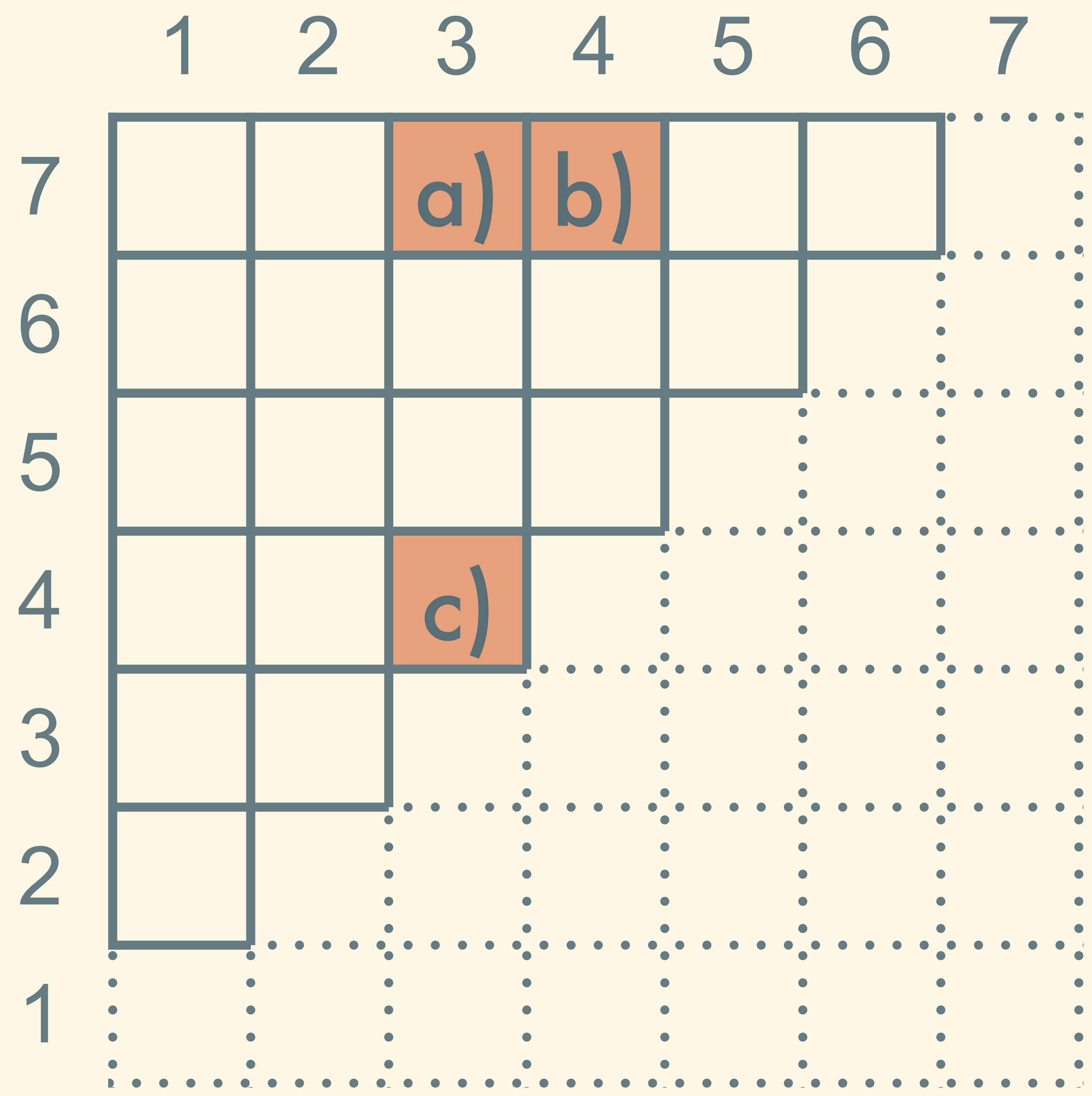


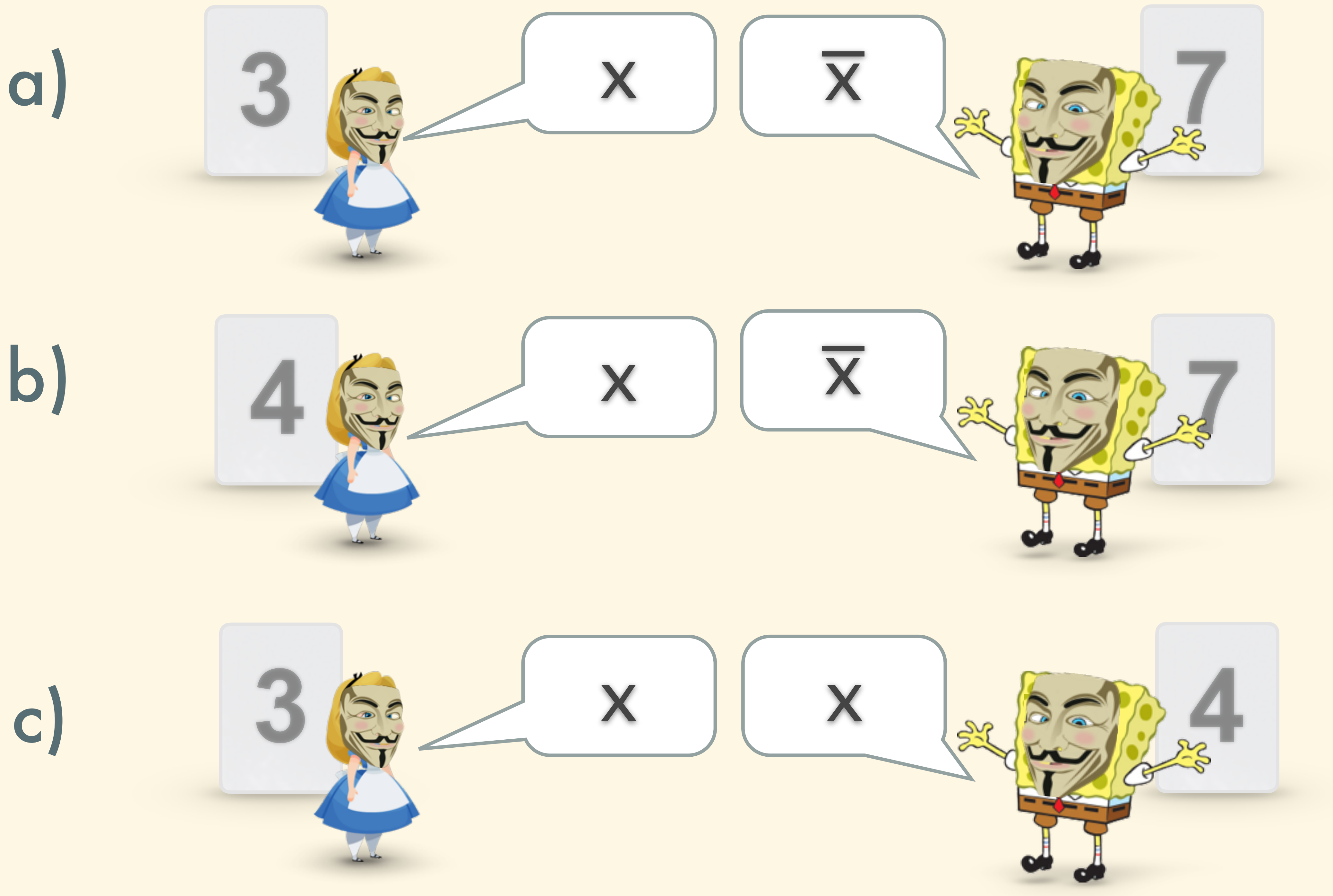


Pair breaking problem

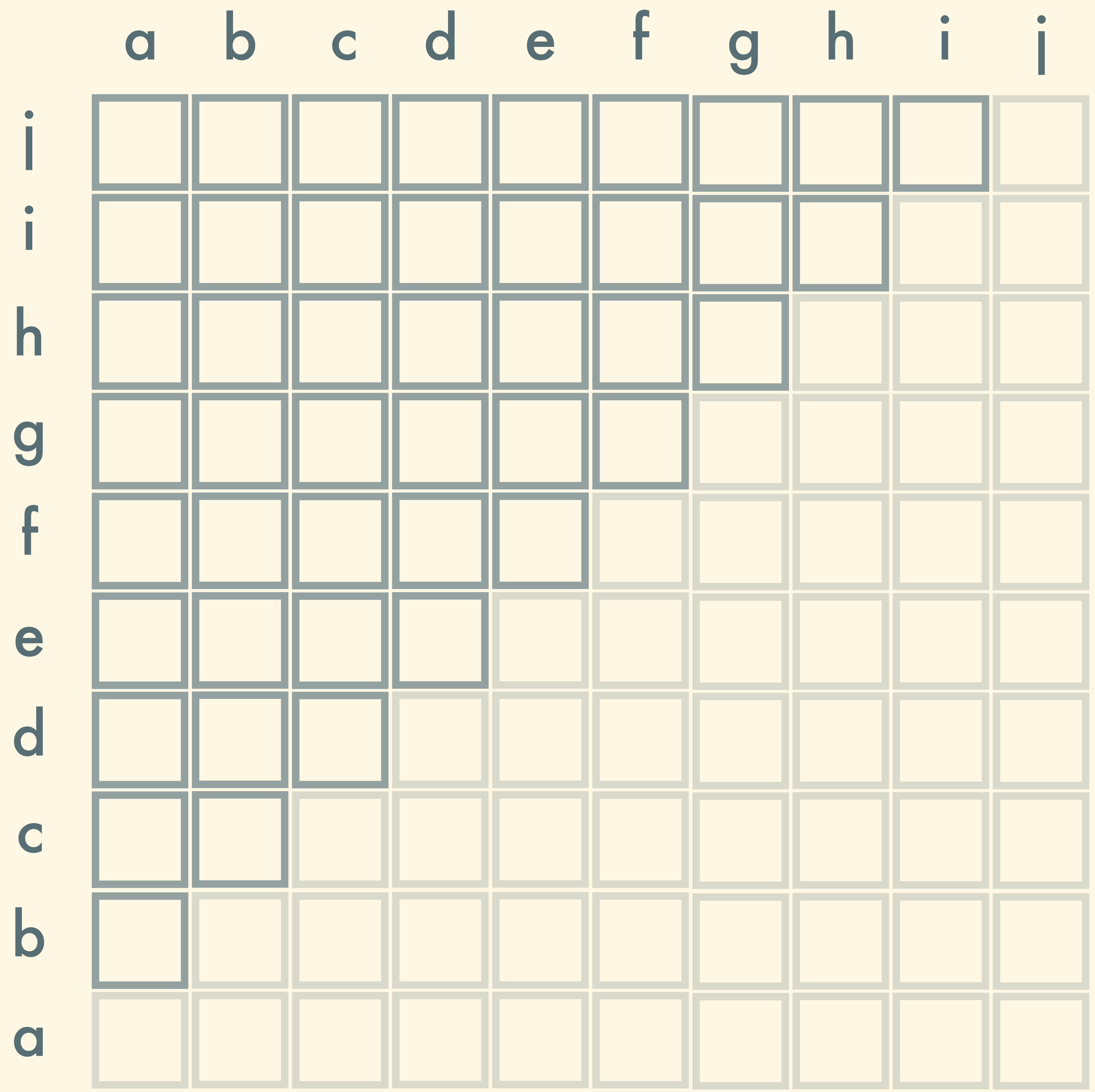
Coloration rules

Invalid coloration: no algorithm can solve the problem if this pattern appears.

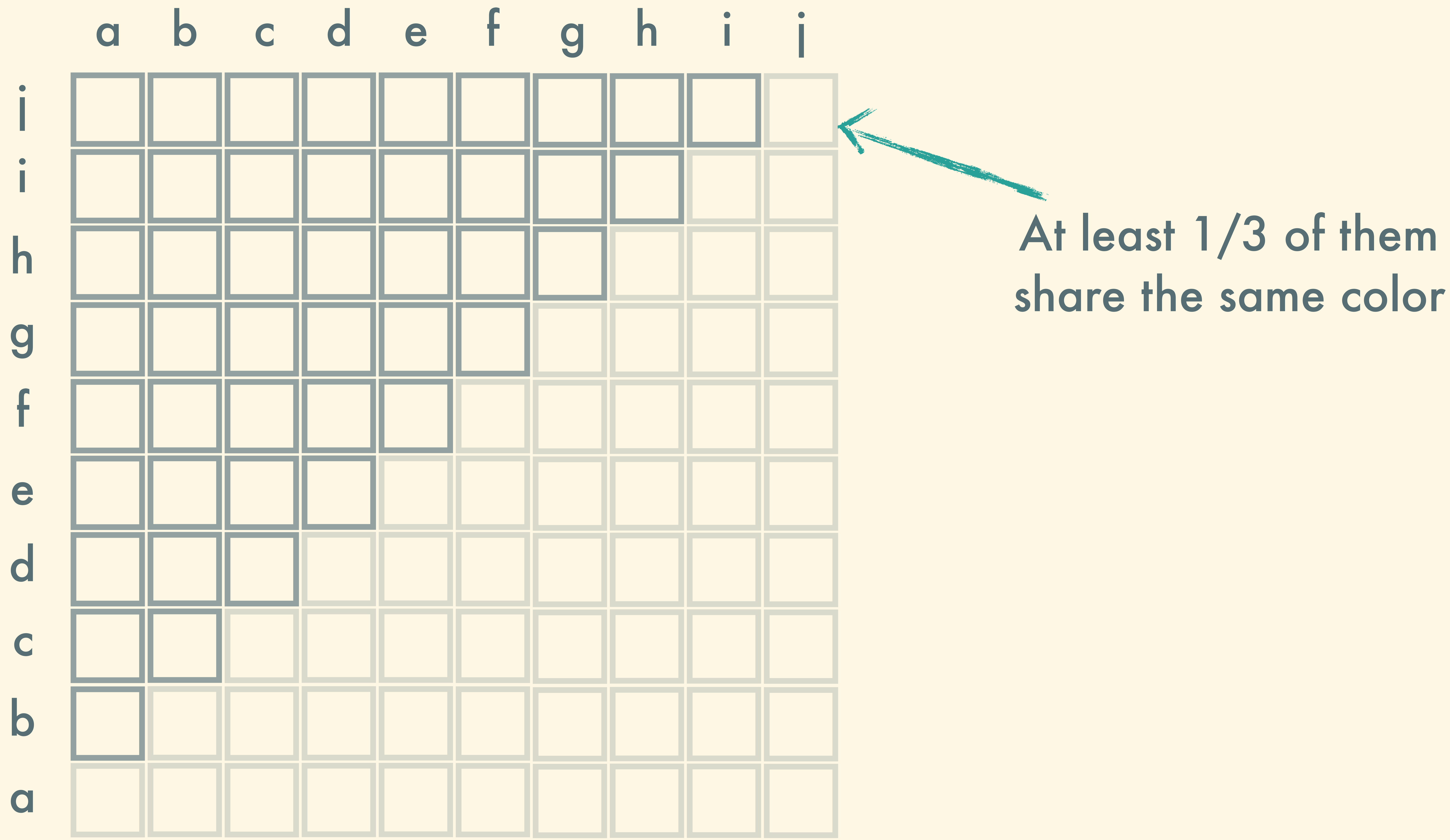




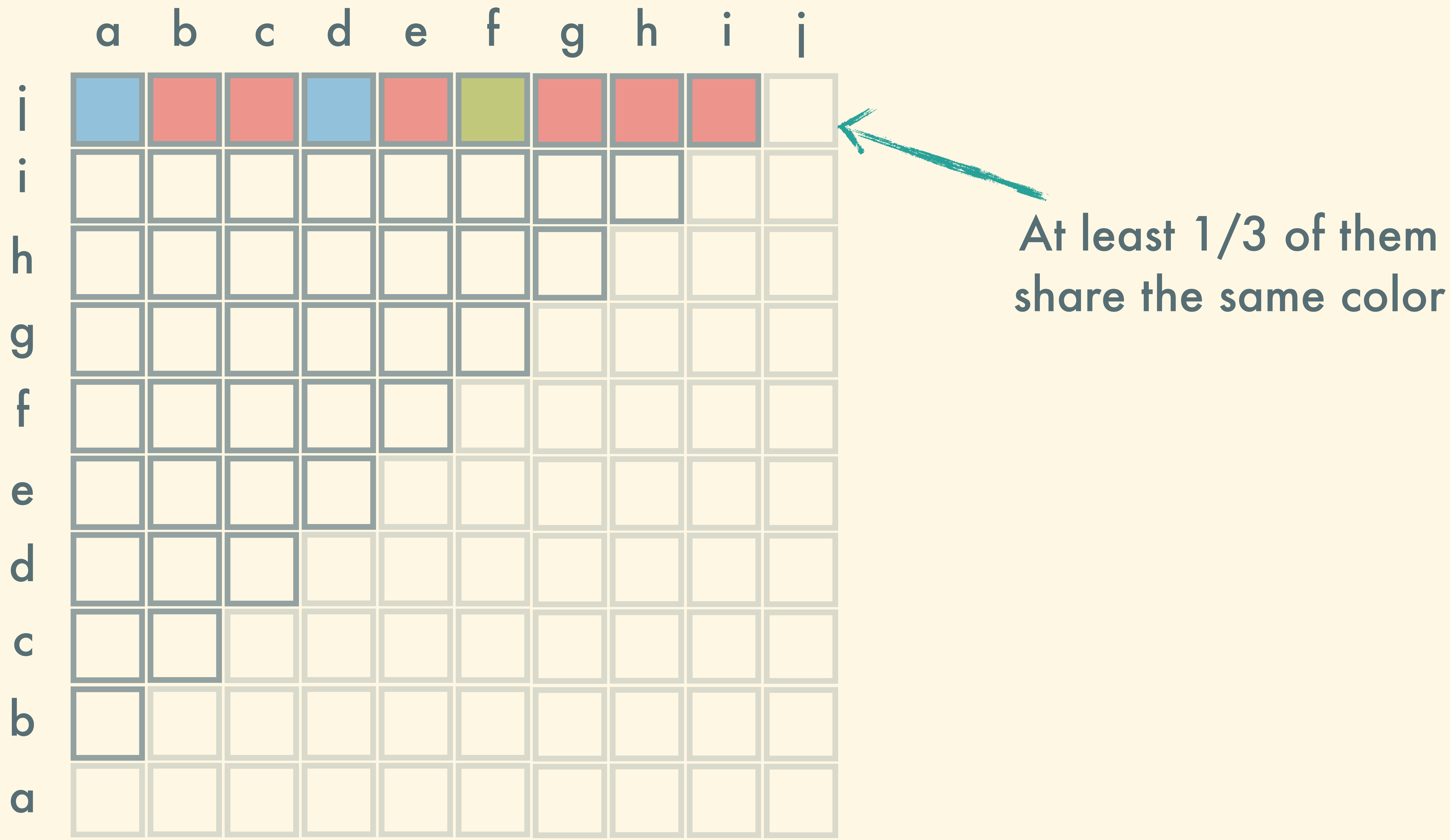
3 colors available



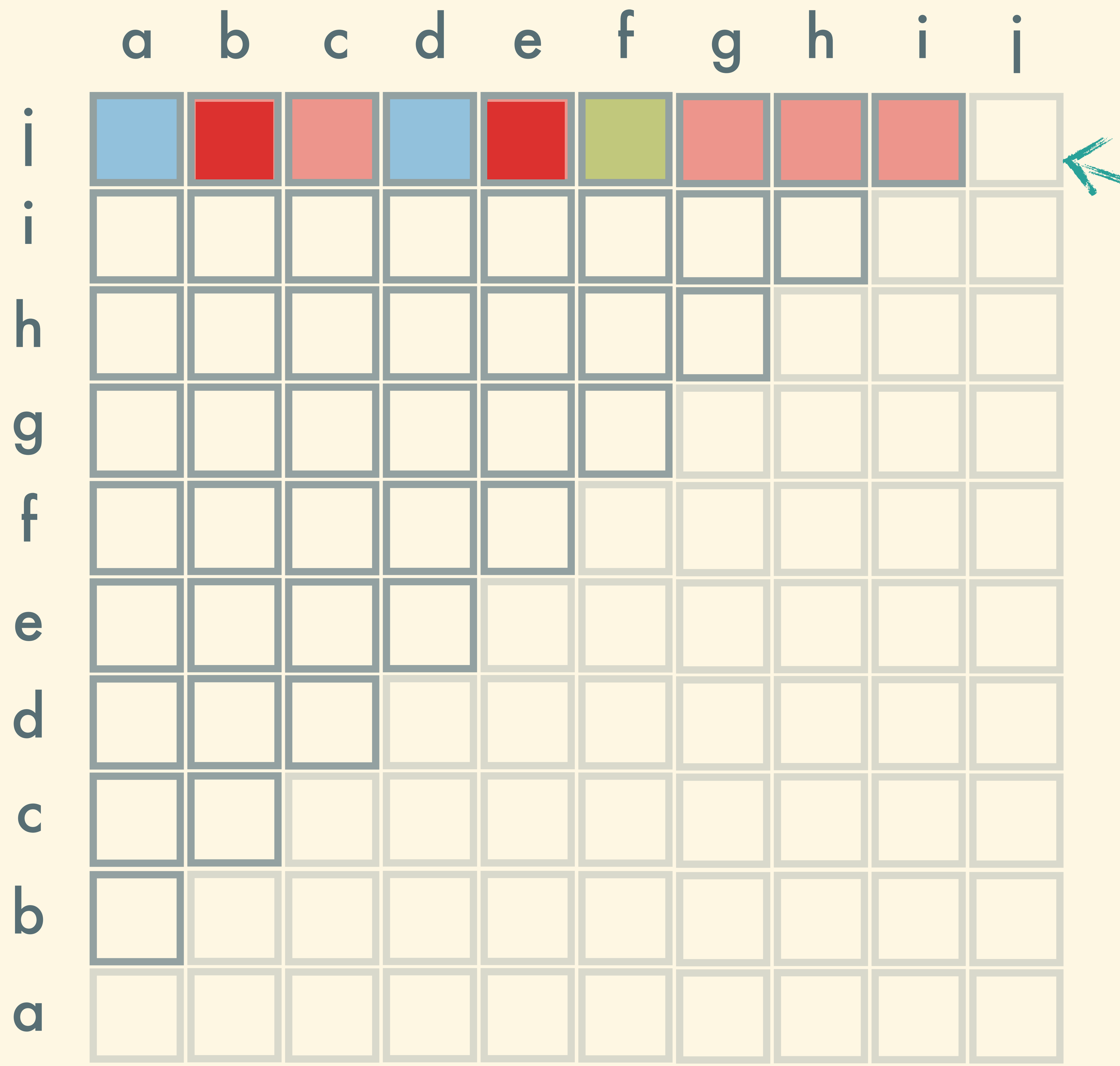
3 colors available



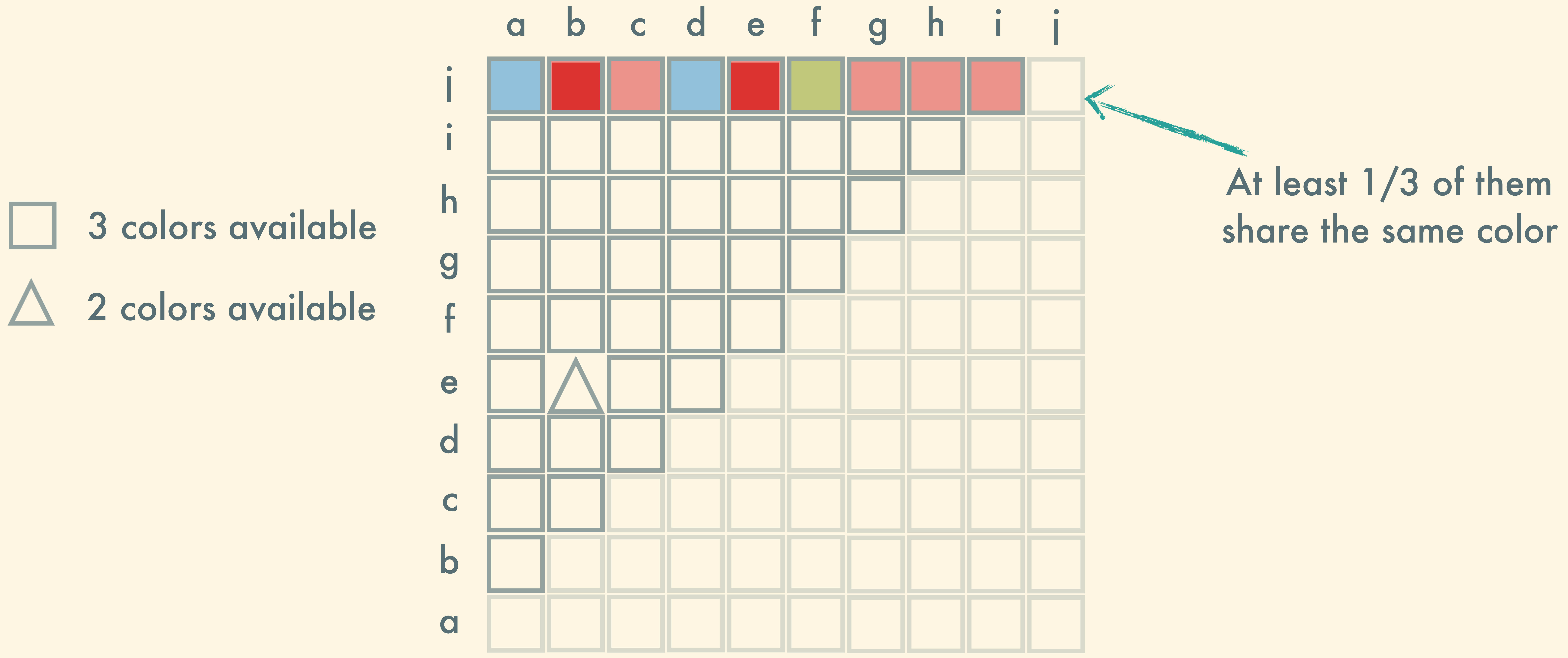
3 colors available

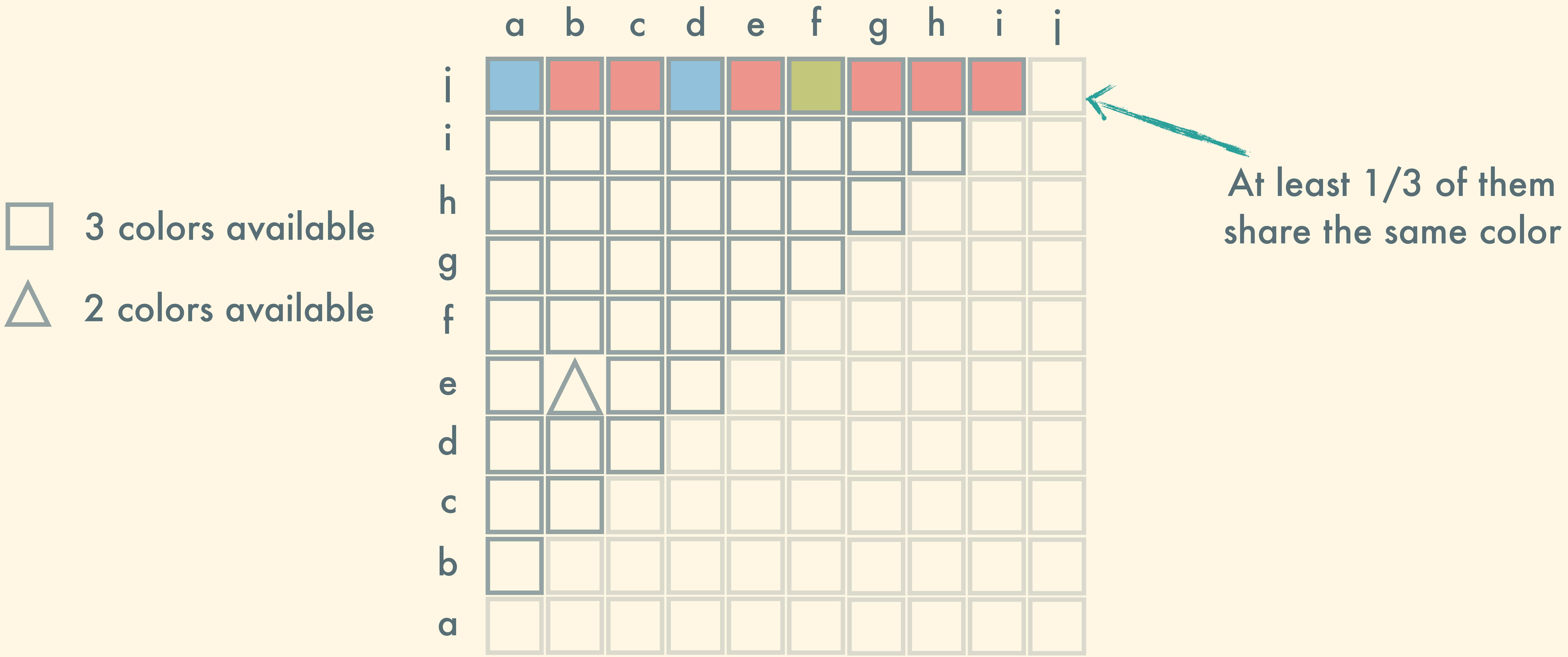


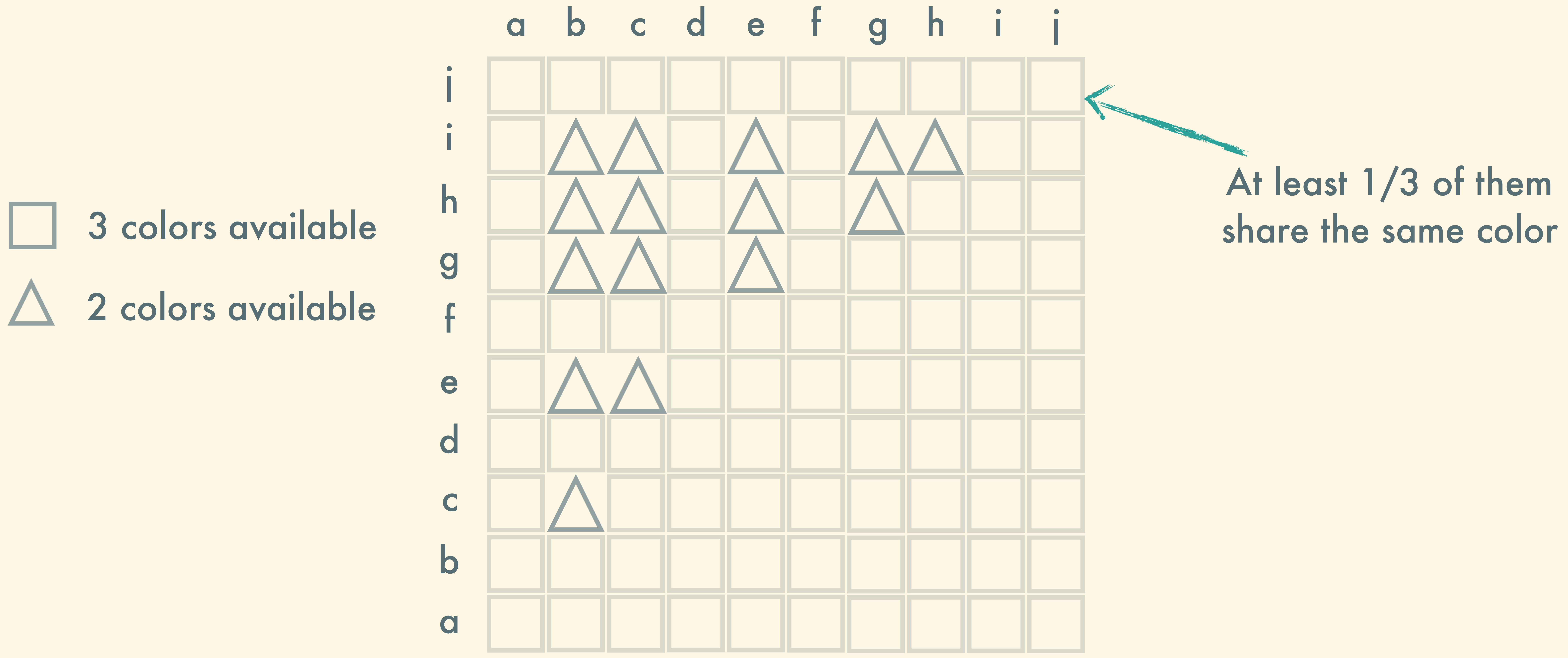
3 colors available

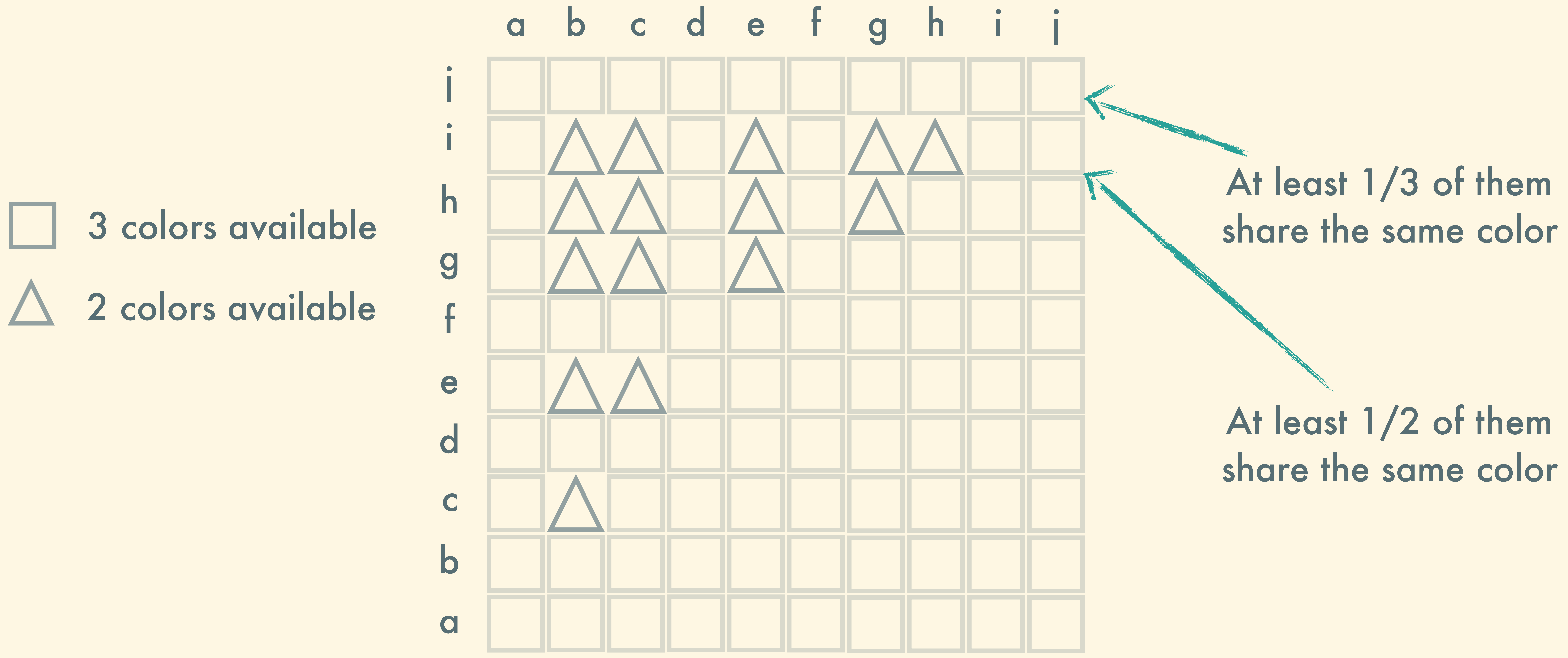


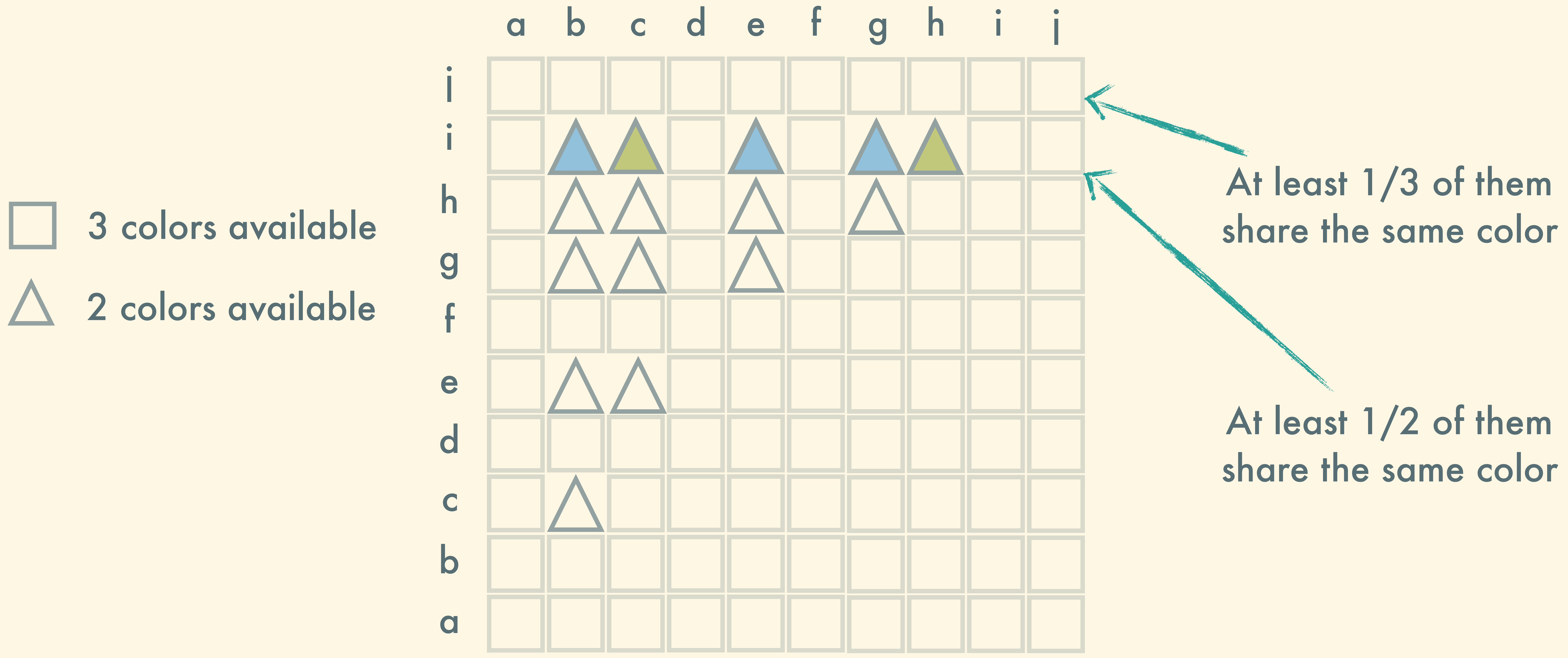
At least 1/3 of them share the same color

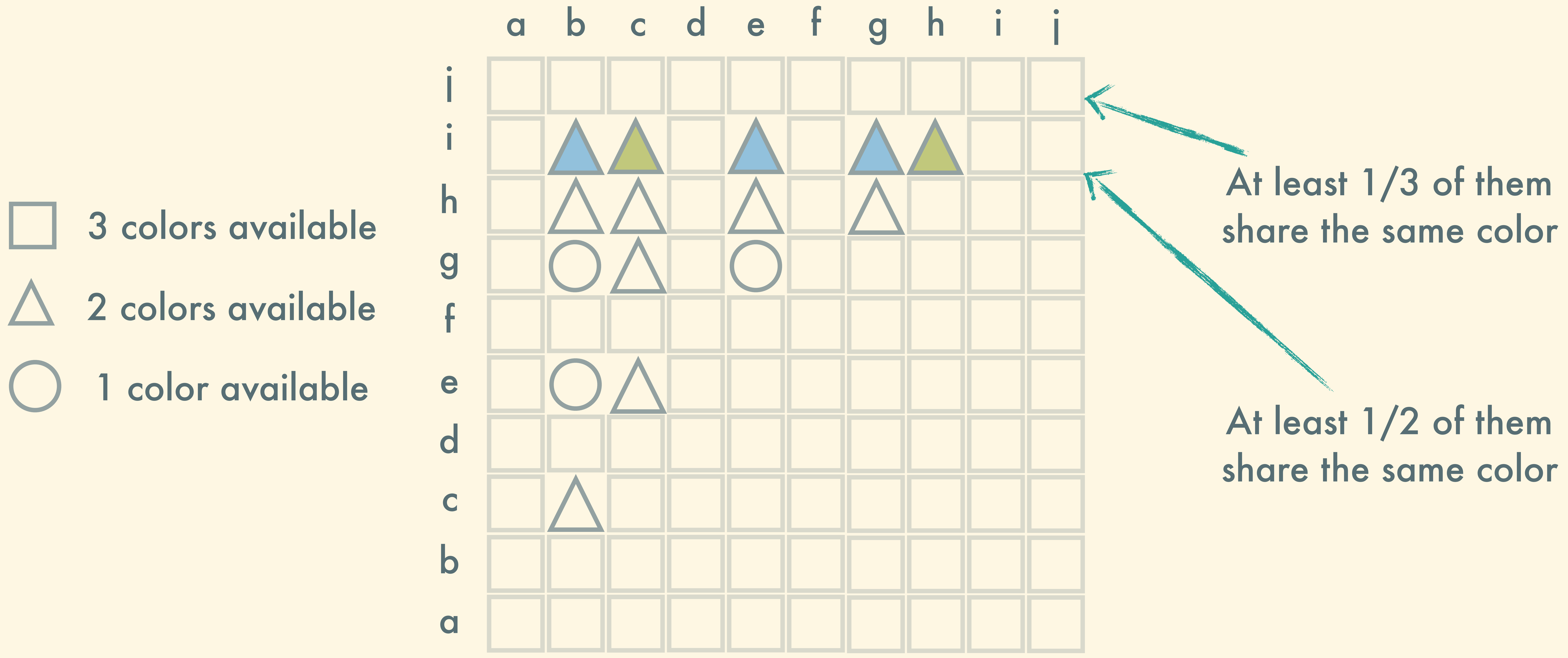

3 colors available
2 colors available
a b c d e f g h i j
j i h g f e d c b a
At least 1/3 of them share the same color

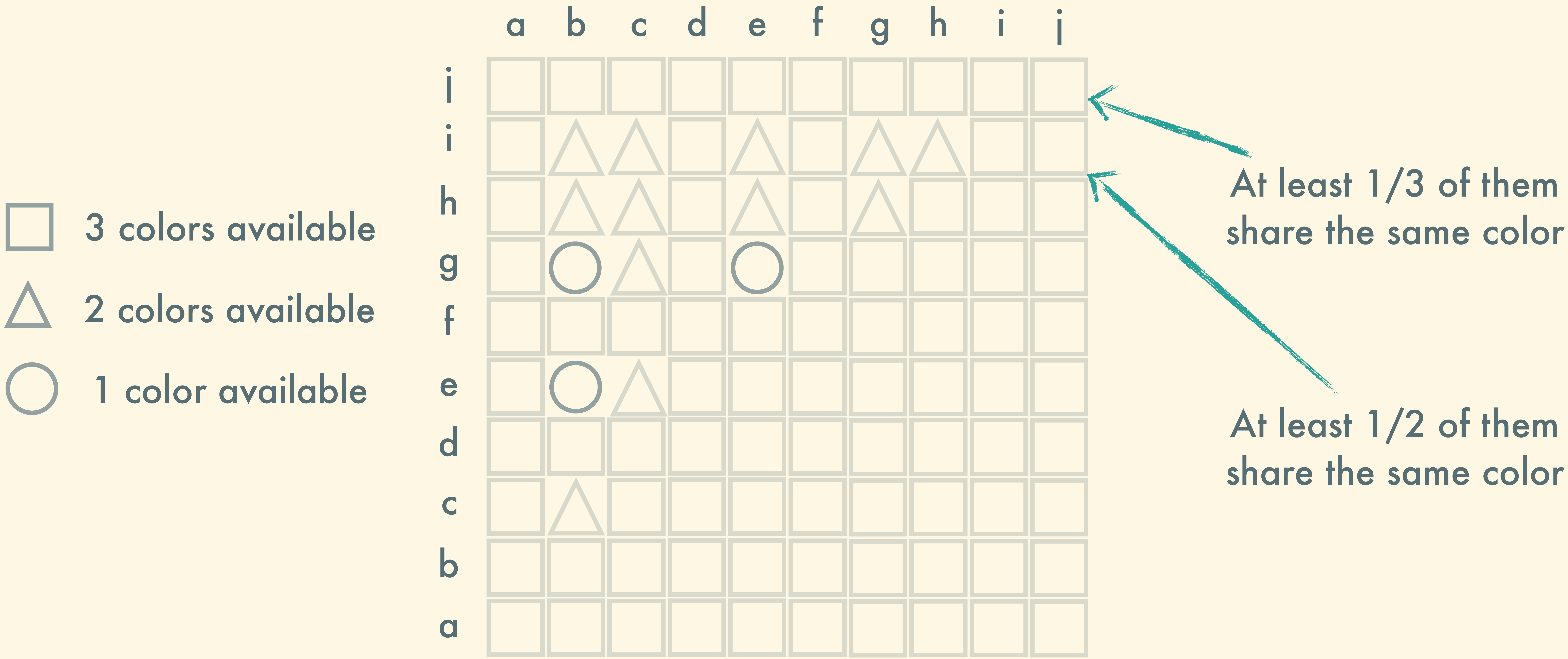

3 colors available
2 colors available
a b c d e f g h i j
j i h g f e d c b a
At least 1/3 of them share the same color

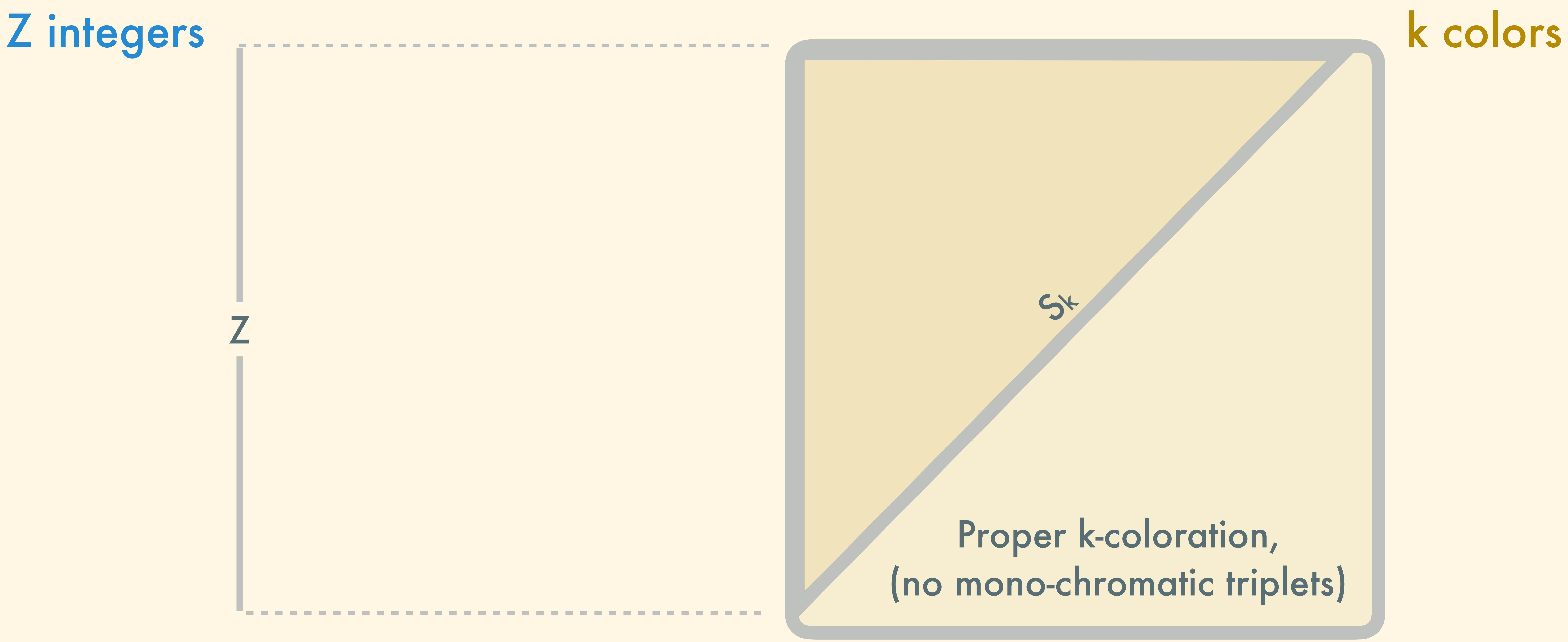

a b c d e f g h i j
j
i
h
g
f
e
d
c
b
a
3 colors available
2 colors available
At least 1/3 of them share the same color

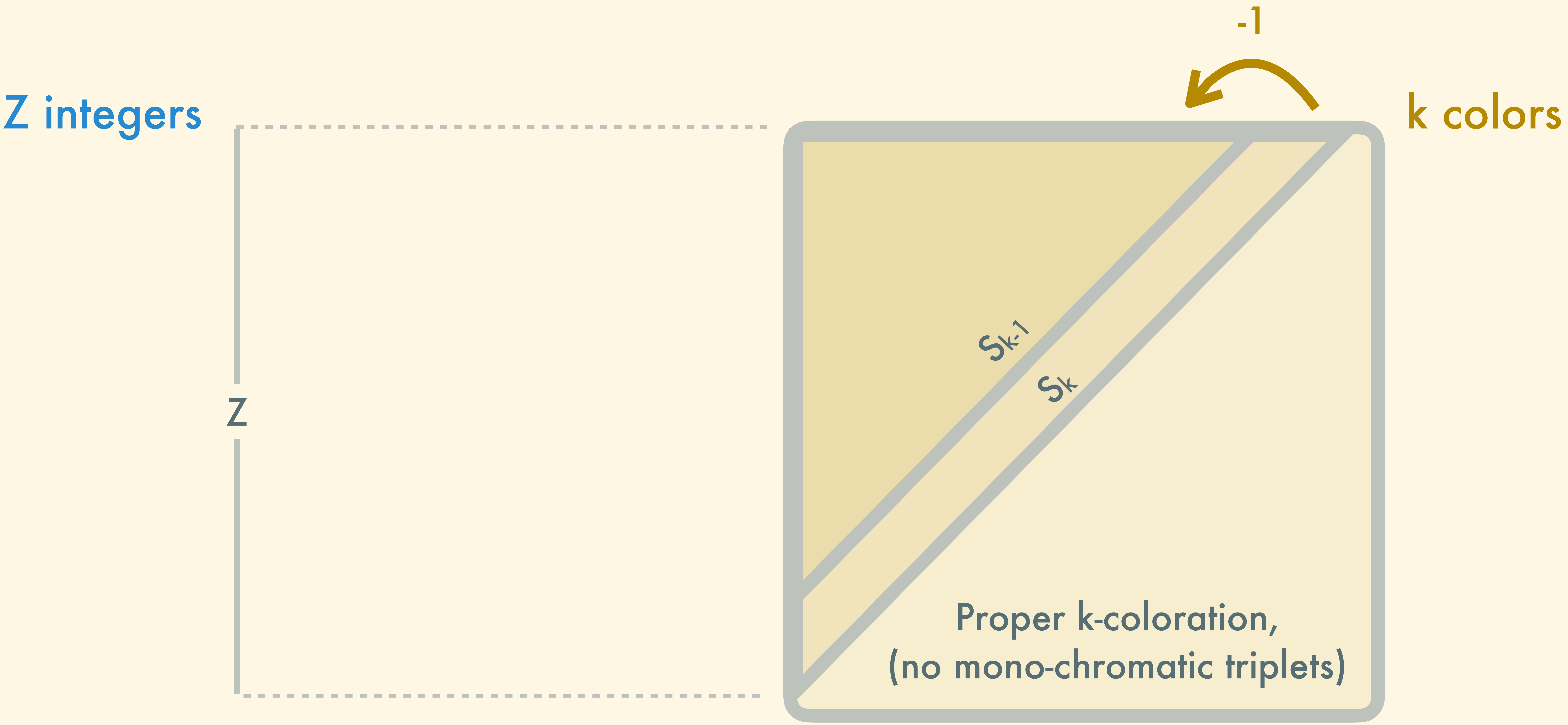

3 colors available
2 colors available
a b c d e f g h i j
j i h g f e d c b a
At least 1/3 of them share the same color
At least 1/2 of them share the same color

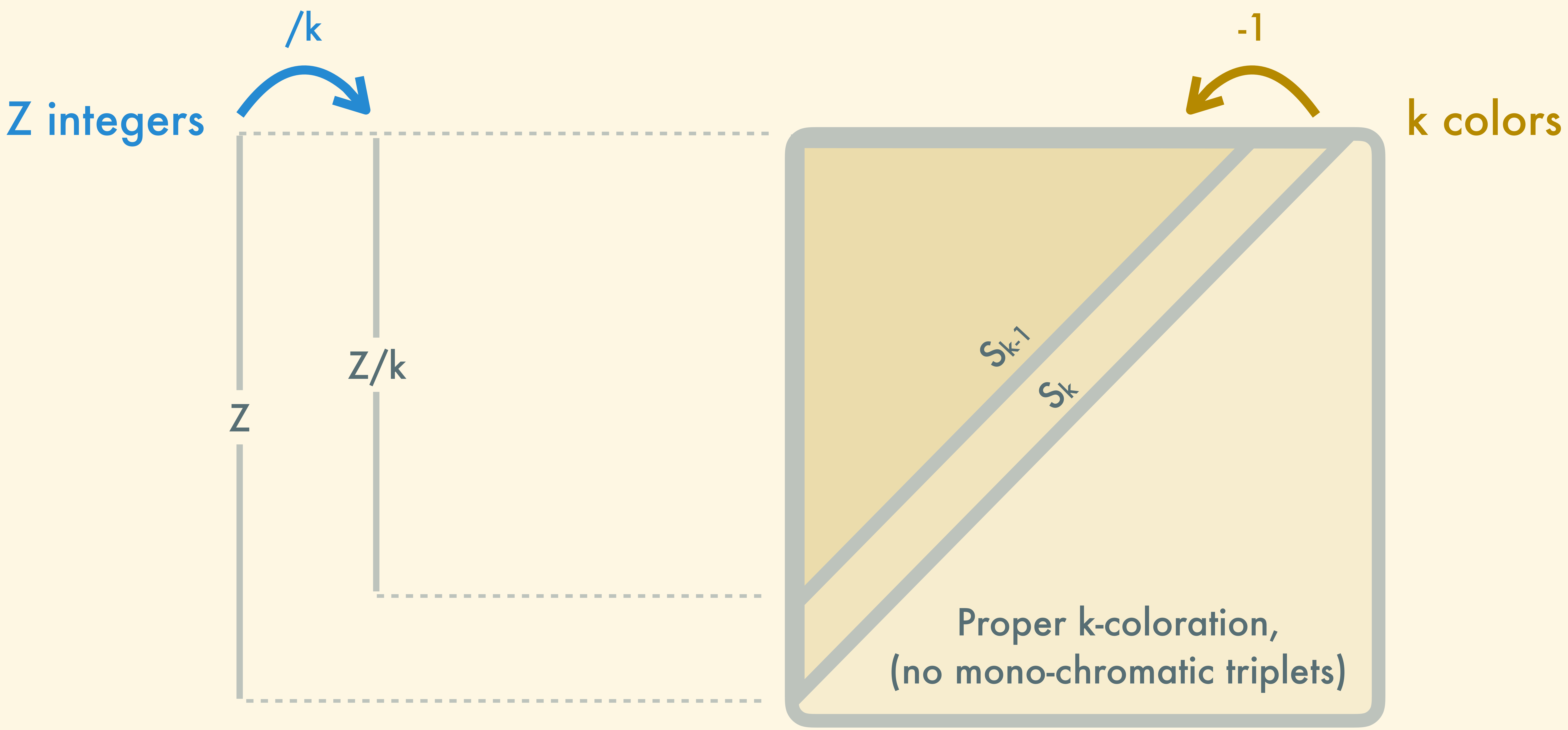

3 colors available
2 colors available
a b c d e f g h i j
j i h g f e d c b a
At least 1/3 of them share the same color
At least 1/2 of them share the same color

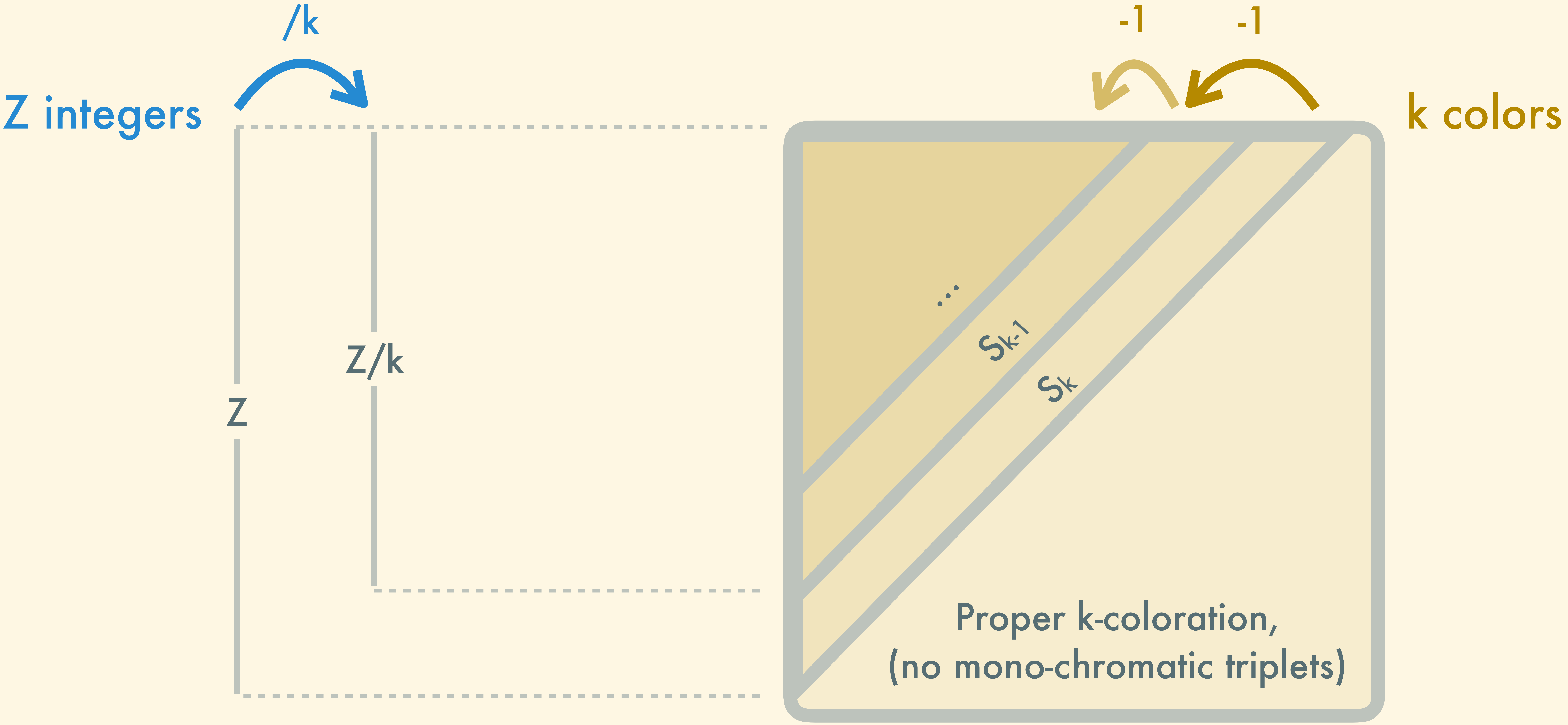

3 colors available
2 colors available
1 color available
a b c d e f g h i j
j i h g f e d c b a
At least 1/3 of them share the same color
At least 1/2 of them share the same color

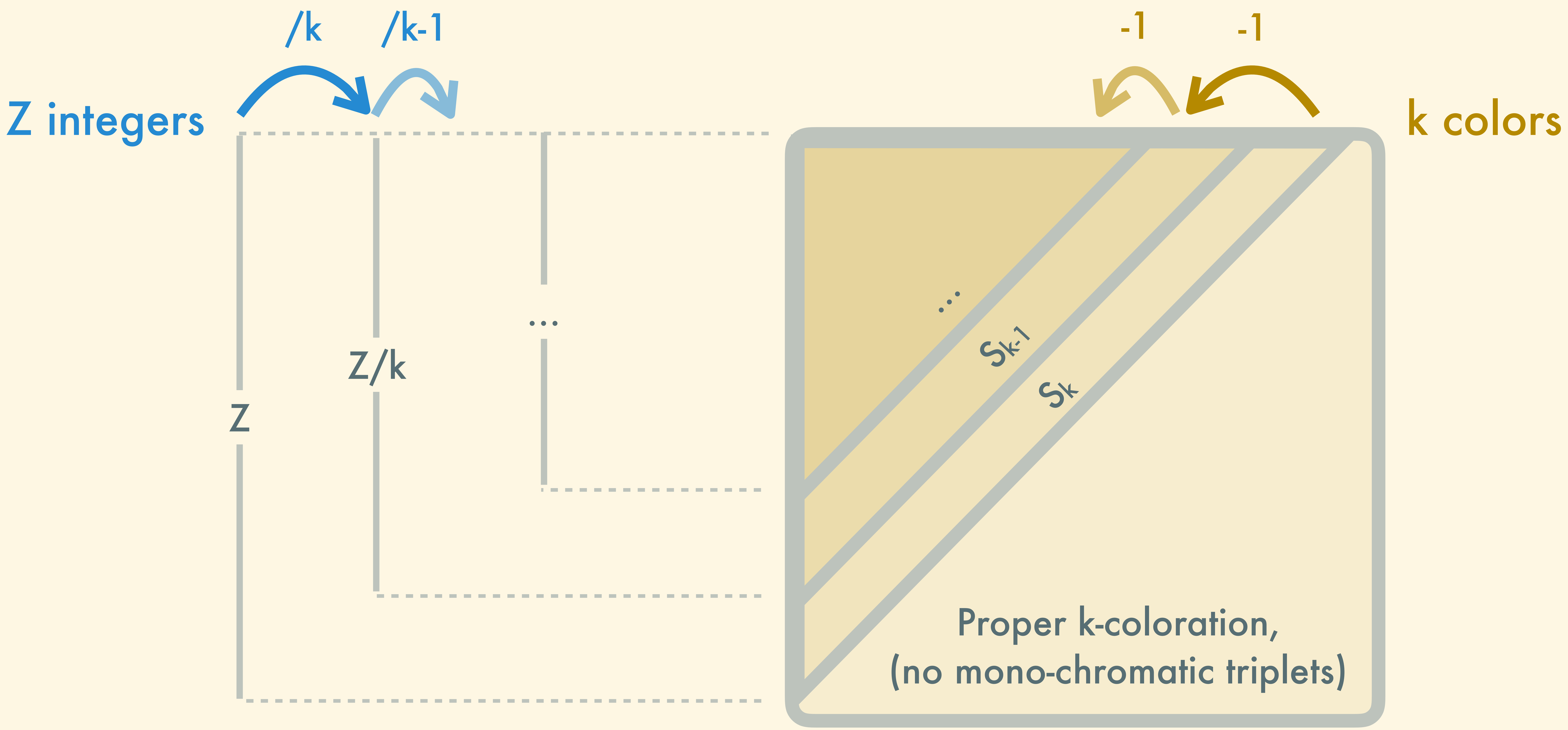

3 colors available
2 colors available
1 color available
a b c d e f g h i j
j i h g f e d c b a
At least 1/3 of them share the same color
At least 1/2 of them share the same color

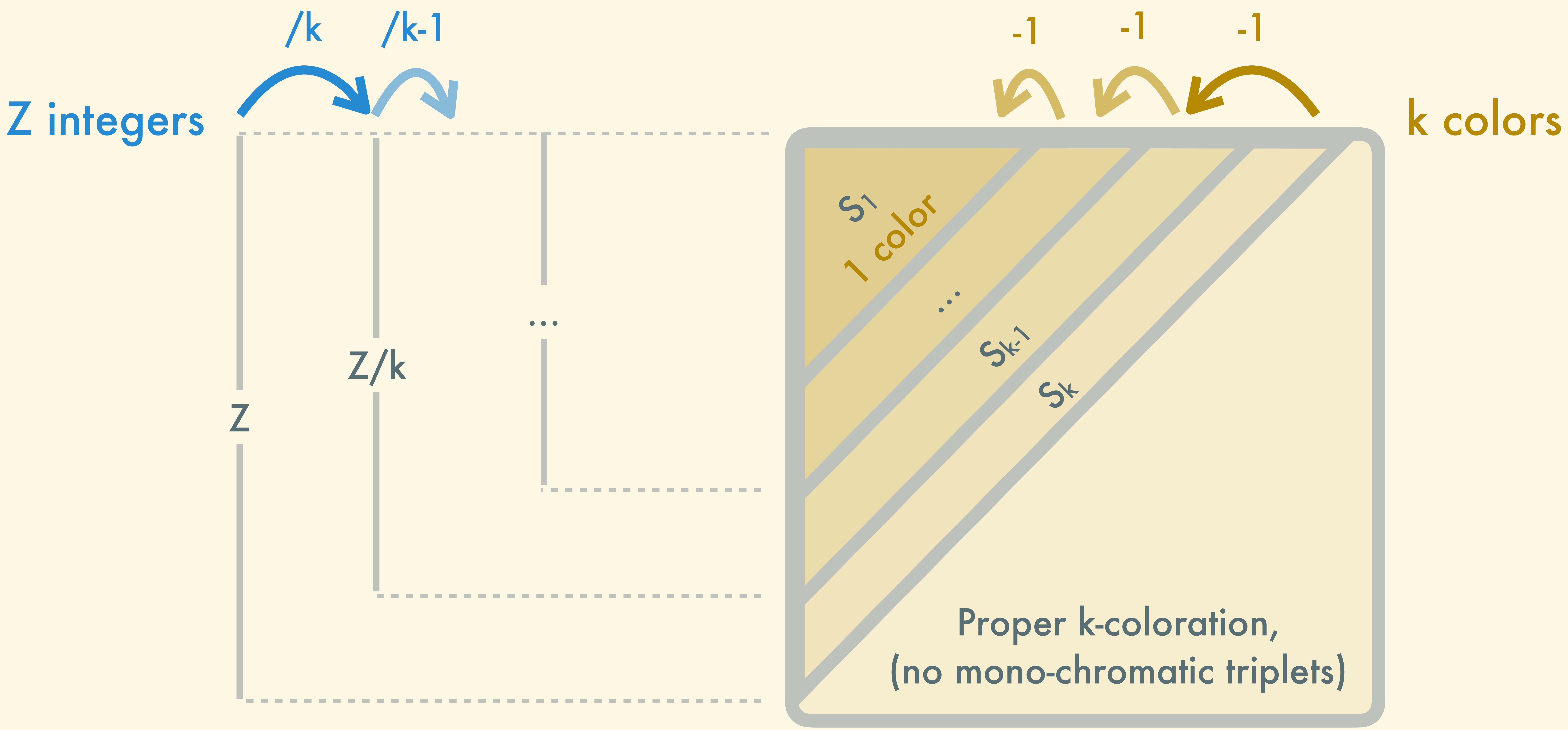

Z integers
k colors
Z
S_k
Proper k-coloration,
(no mono-chromatic triplets)

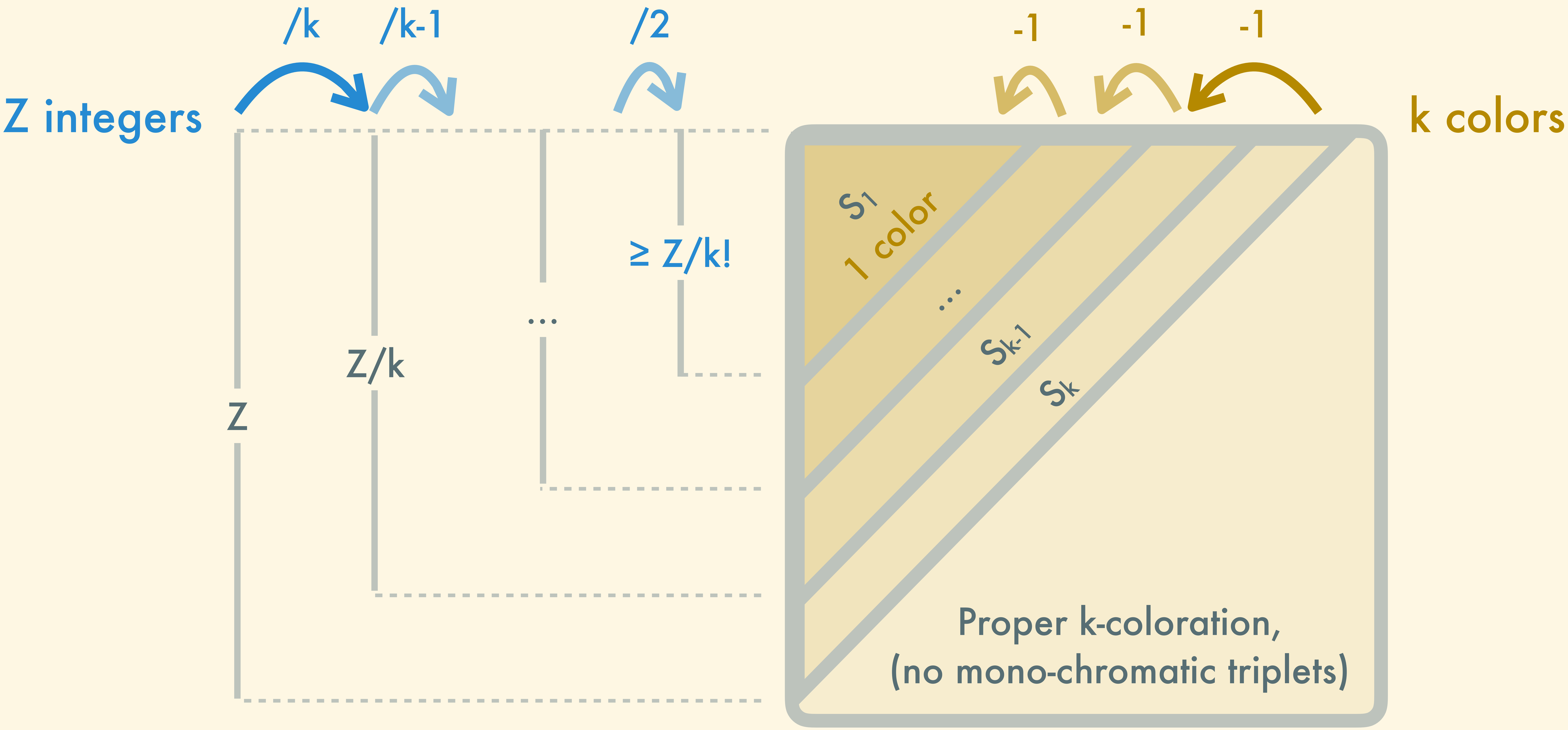

Z integers
k colors
-1
Z
S_{k-1}
S_k
Proper k-coloration,
(no mono-chromatic triplets)


/k
-1
Z integers
k colors
Z
Z/k
Sk-1
Sk
Proper k-coloration,
(no mono-chromatic triplets)


/k
Z integers
Z/k
Z
-1
-1
k colors
...
Sk-1
Sk
Proper k-coloration,
(no mono-chromatic triplets)


/k
/k-1
Z integers
-1
-1
k colors
Z
Z/k
...
...
S_{k-1}
S_k
Proper k-coloration,
(no mono-chromatic triplets)


/k
/k-1
-1
-1
-1
Z integers
k colors
Z
Z/k
...
S1
1 color
...
Sk-1
Sk
Proper k-coloration,
(no mono-chromatic triplets)



/k
/k-1
/2
-1
-1
-1
Z integers
k colors
Z
Z/k
...
≥ Z/k!
S1
1 color
...
Sk-1
Sk
Proper k-coloration,
(no mono-chromatic triplets)

Pair breaking game

Pair breaking game

- Remember that if $|S_1| \geq Z/k! \geq 3$ then S_1 is not coloured properly

Pair breaking game

- Remember that if $|S_1| \geq Z/k! \geq 3$ then S_1 is not coloured properly
- Thus, if S_k is coloured properly with k colors then $Z/k^k \leq Z/k! < 3$

PAIR BREAKING GAME

$Z < 3k^k$

$\log Z < 3k*\log(k)$

$\log Z < 3k^2$

- Remember that if $|S_1| \geq Z/k! \geq 3$ then S_1 is not coloured properly
- Thus, if S_k is coloured properly with k colors then $Z/k^k \leq Z/k! < 3$

Pair breaking game

$Z < 3k^k$

$\log Z < 3k*\log(k)$

$\log Z < 3k^2$

$k > \sqrt{(1/3*\log Z)}$

- Remember that if $|S_1| \geq Z/k! \geq 3$ then S_1 is not coloured properly
- Thus, if S_k is coloured properly with k colors then $Z/k^k \leq Z/k! < 3$
- $\Rightarrow k > \sqrt{(1/3*\log Z)}$

Pair breaking game

$Z < 3k^k$

$\log Z < 3k*\log(k)$

$\log Z < 3k^2$

$k > \sqrt{(1/3*\log Z)}$

Number of bits to encode these colors:

$> \log(\sqrt{(1/3*\log Z)})$

$>1/2(\log (1/3*\log Z))$

- Remember that if $|S_1| \geq Z/k! \geq 3$ then S_1 is not coloured properly
- Thus, if S_k is coloured properly with k colors then $Z/k^k \leq Z/k! < 3$
- $\Rightarrow k > \sqrt{(1/3*\log Z)}$
- The advice has size $\Omega(\log \log Z)$.

Lower bound

For a time exactly D-2 and D odd

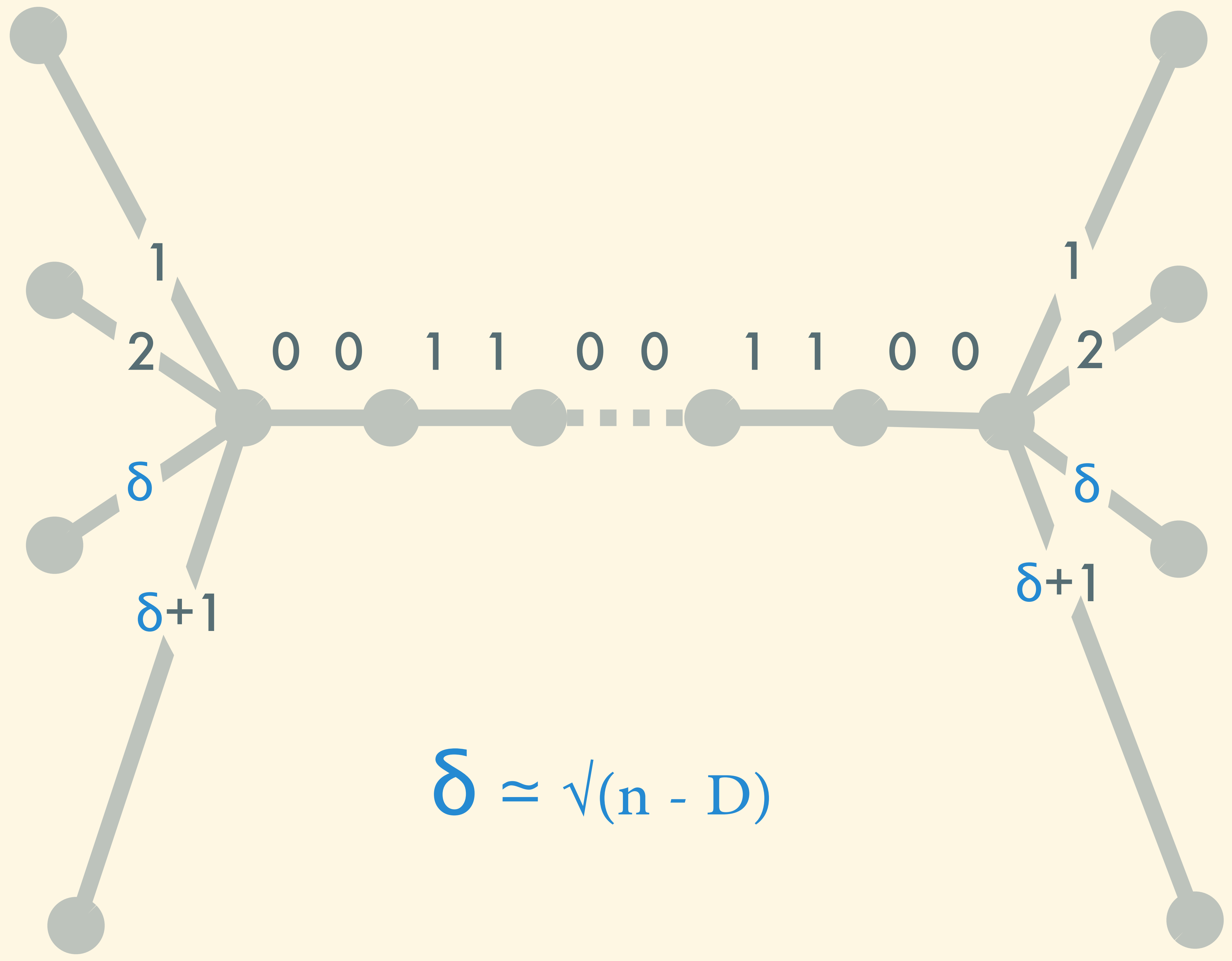
- Reduction to a pair breaking problem:

1. The pair breaking problem requires Ω(log log Z) bits of advice

2. Use an hypothetical algorithm, ELECT, that solves leader election in time D-2 using o(log log Z) bits to solve the pair breaking problem.

Double brooms

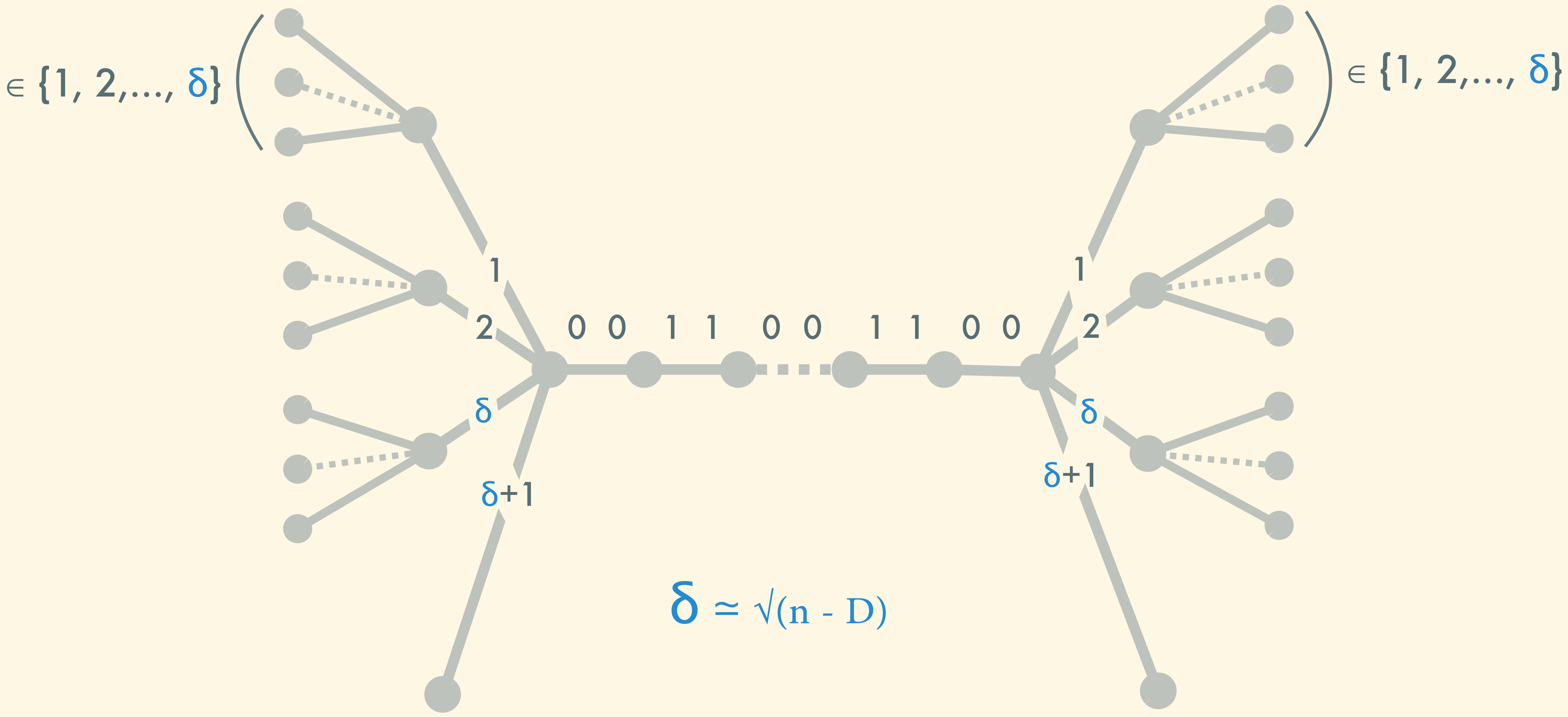
parameter δ



$$\delta \simeq \sqrt{(n - D)}$$

Double brooms

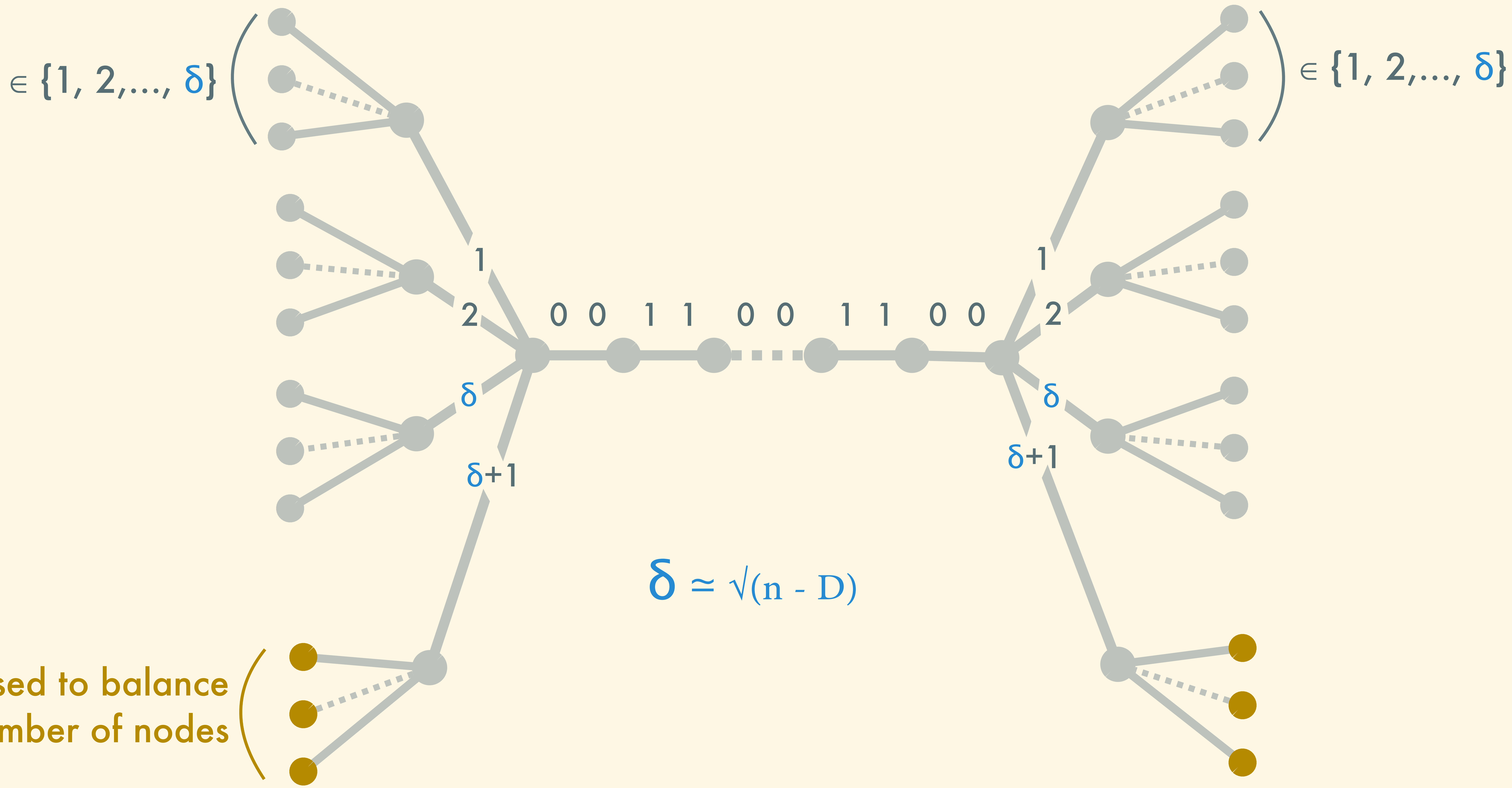
parameter δ



$\delta \simeq \sqrt{(n - D)}$

Double brooms

parameter δ



$\delta \simeq \sqrt{(n - D)}$

Double brooms

parameter δ

The number of leaves is determined by any bijective function f:

$$f: \{1,\ldots,Z\} \rightarrow \{1,\ldots,\delta\}^{\delta}$$

Double brooms

parameter δ

a $f(a)=(a_1,a_2,\ldots,a_\delta)$

b $f(b)=(b_1,b_2,\ldots,b_\delta)$

The number of leaves is determined by any bijective function f:

$f: \{1,\ldots,Z\} \rightarrow \{1,\ldots,\delta\}^\delta$

Double brooms

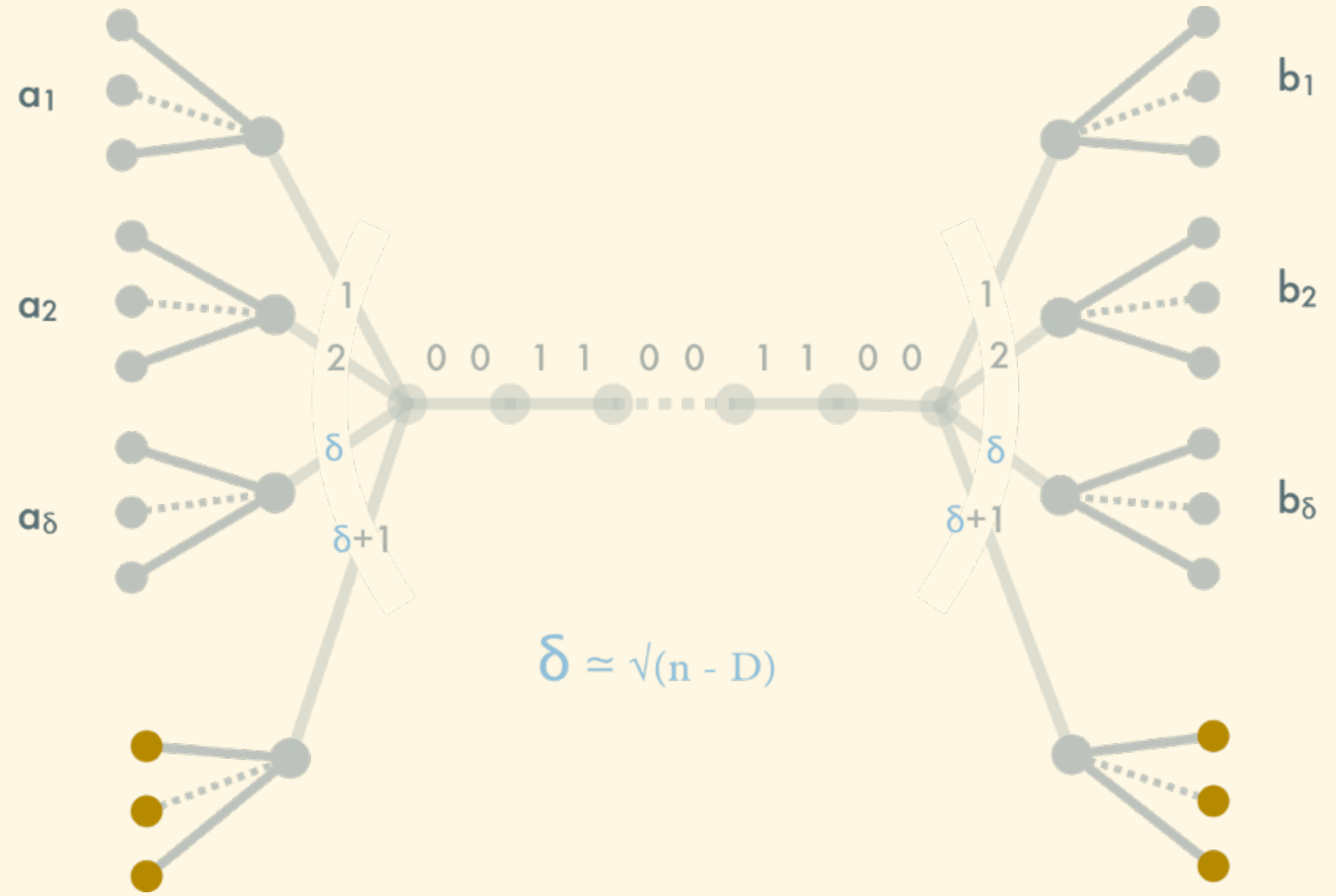
parameter δ

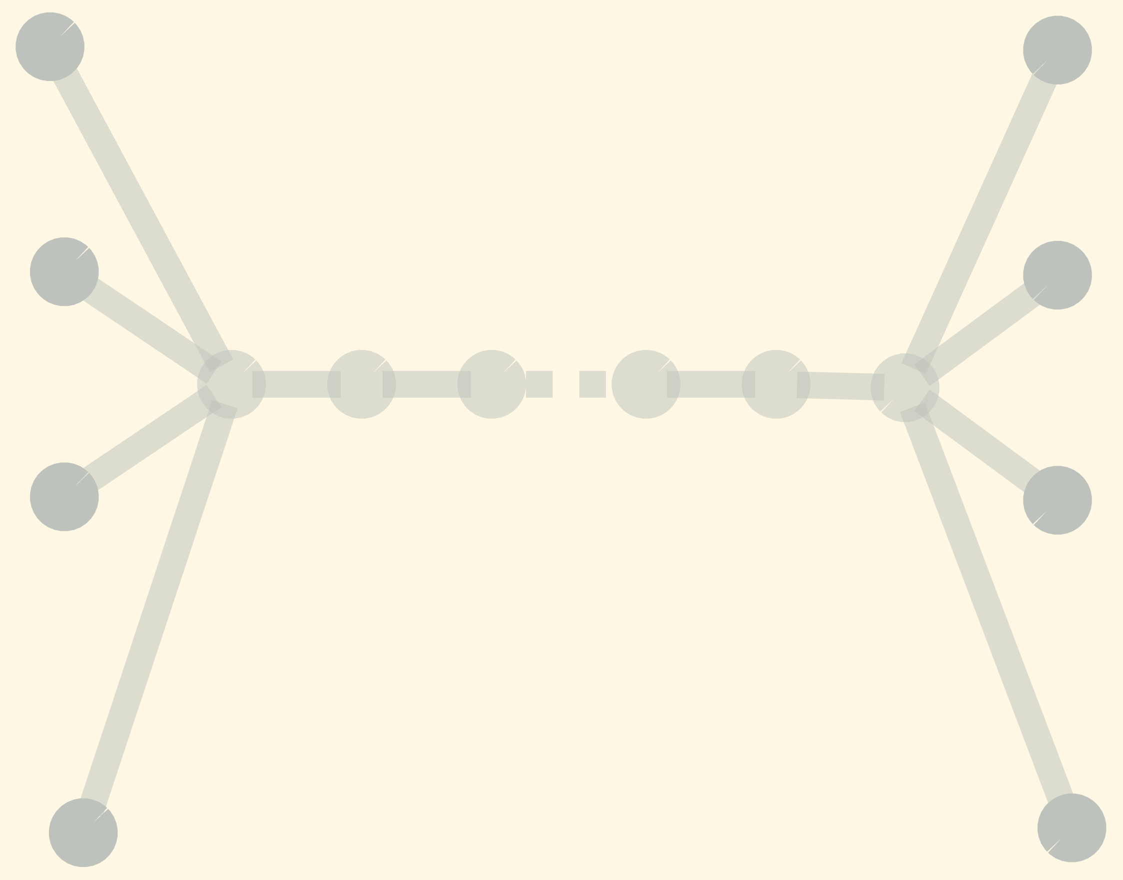
a $f(a)=(a_1,a_2,\ldots,a_\delta)$

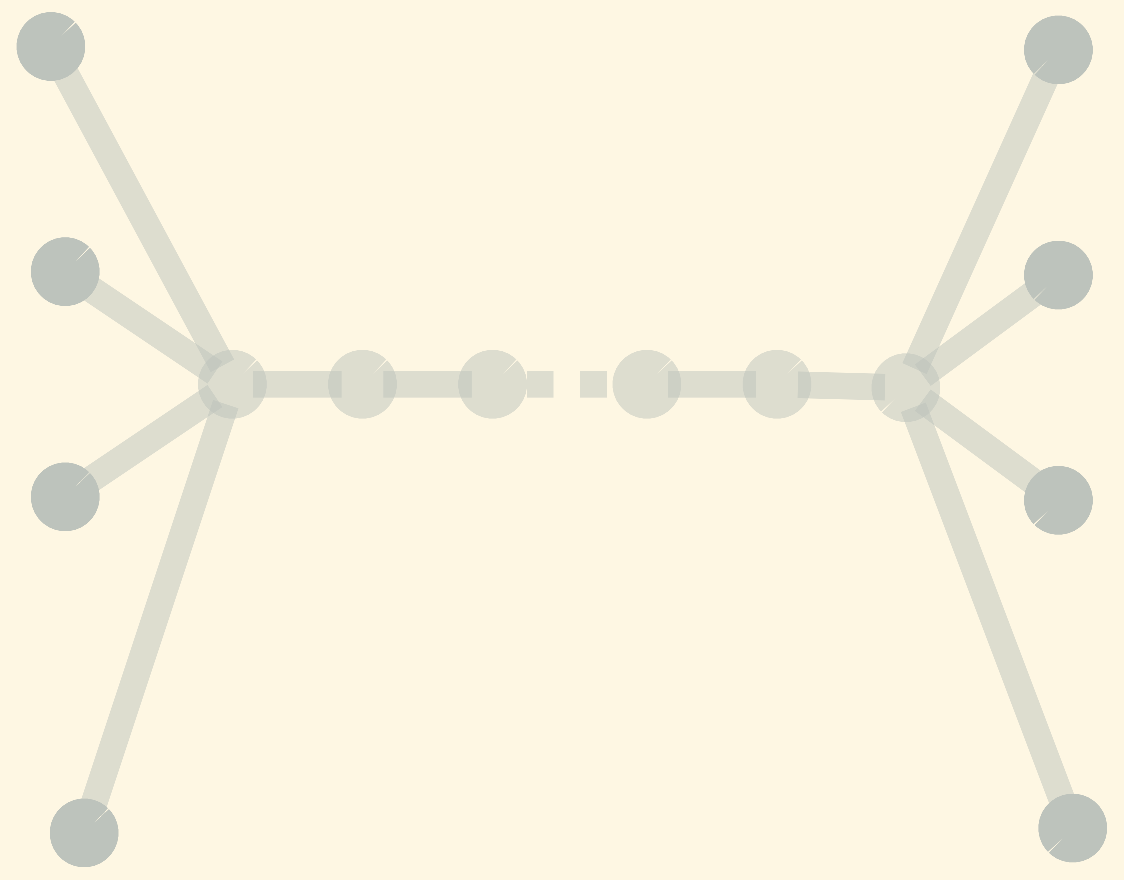
b $f(b)=(b_1,b_2,\ldots,b_\delta)$

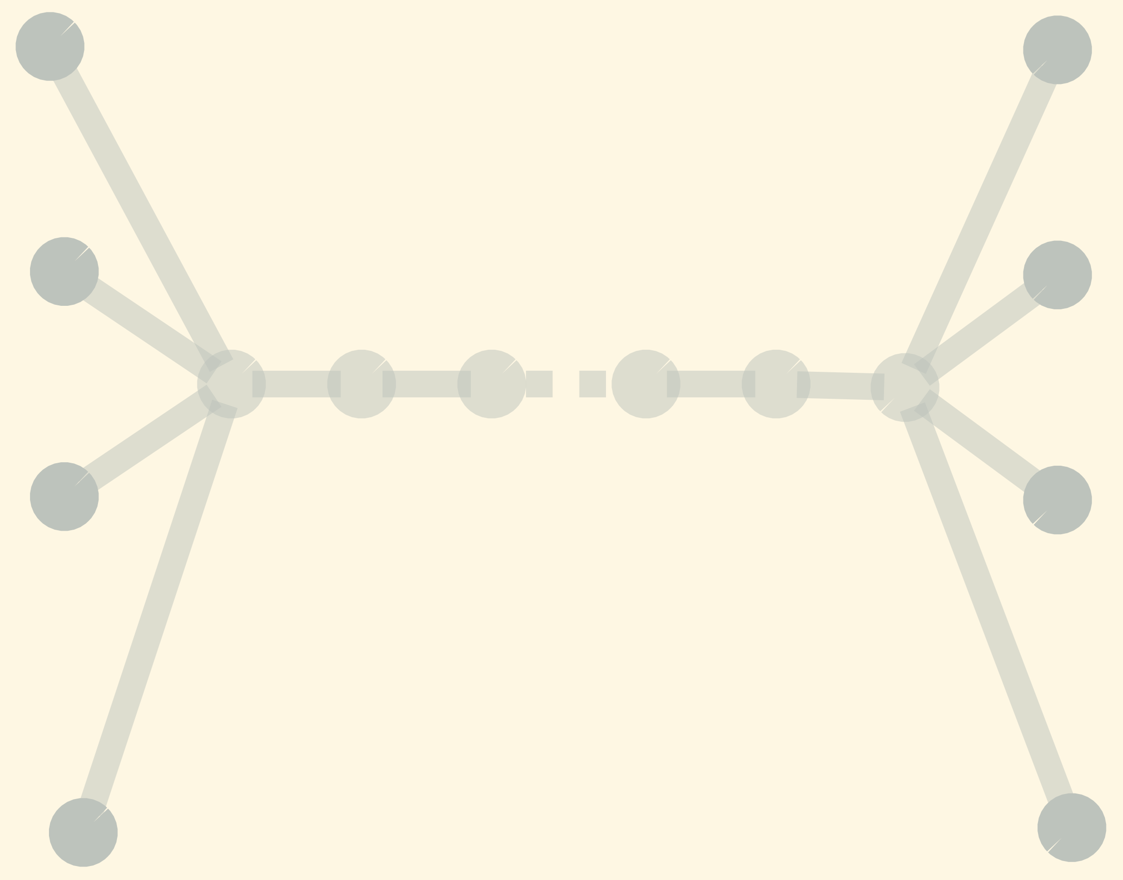
The number of leaves is determined by any bijective function f:

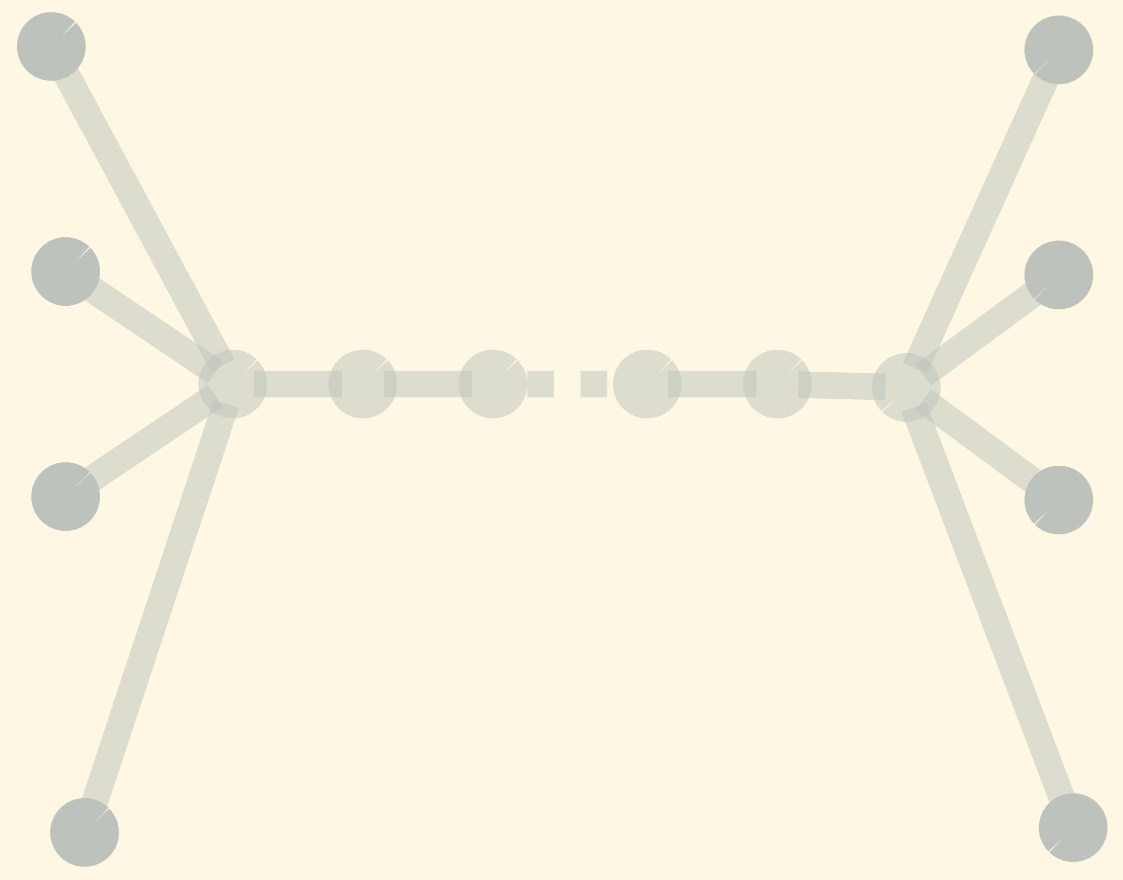
$$f: \{1,\ldots,Z\} \rightarrow \{1,\ldots,\delta\}^{\delta}$$

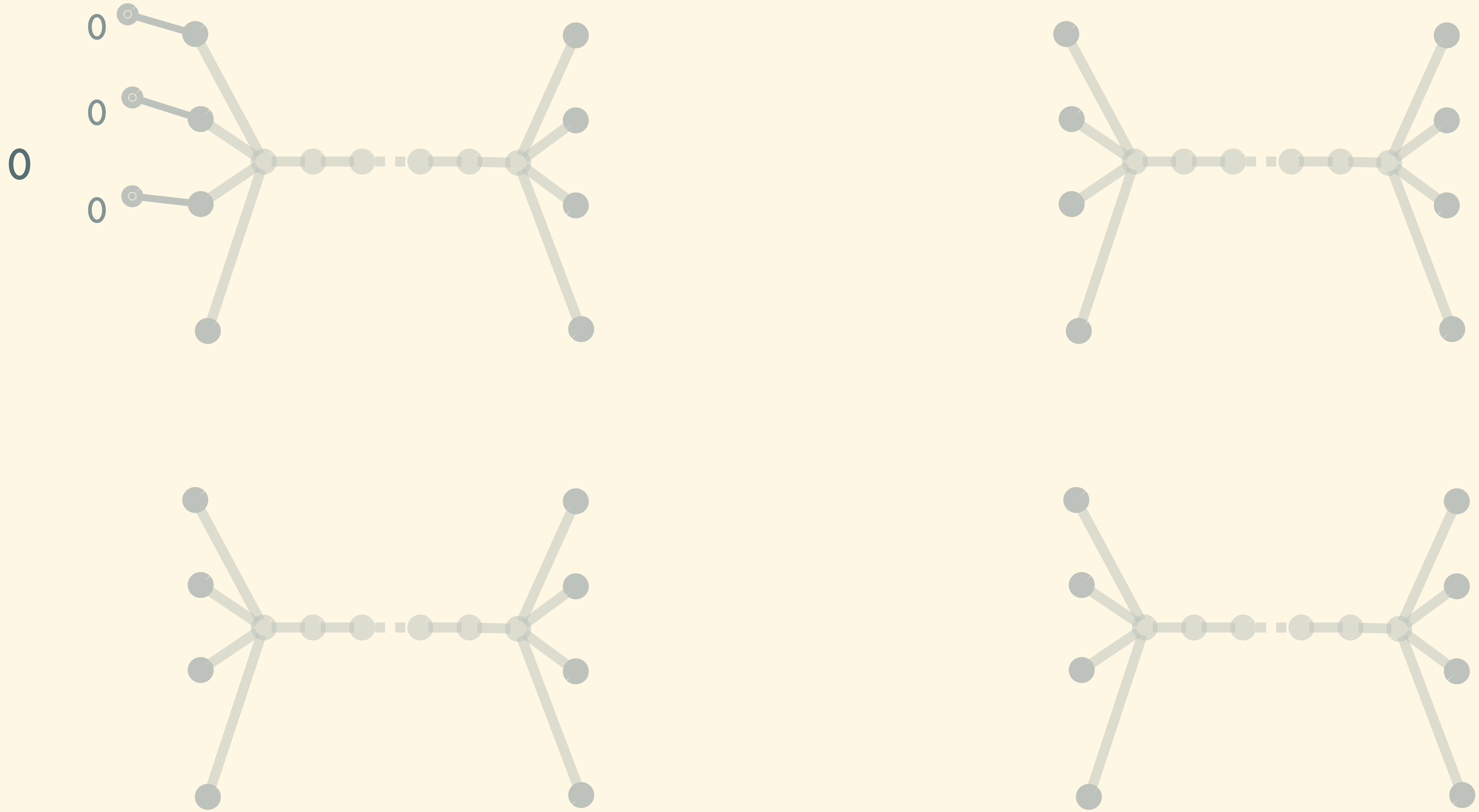


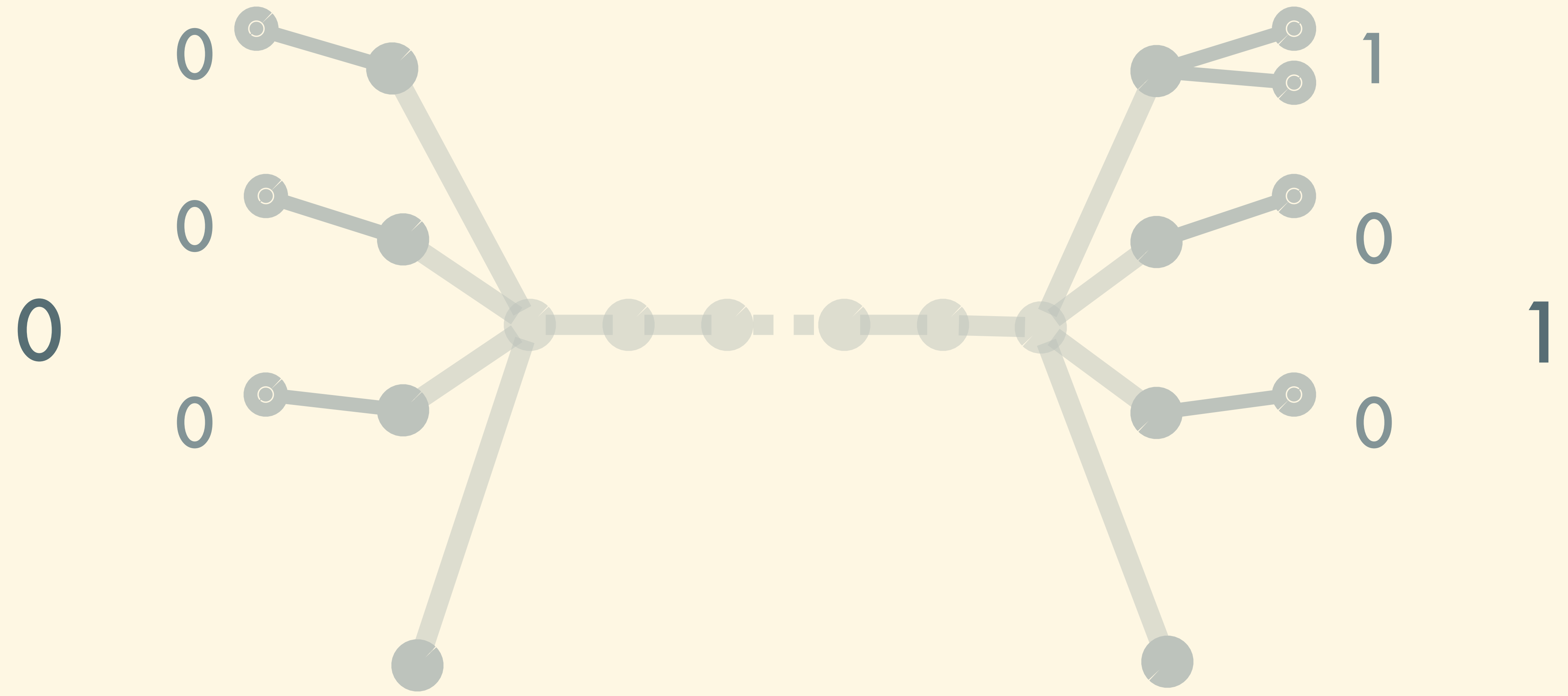


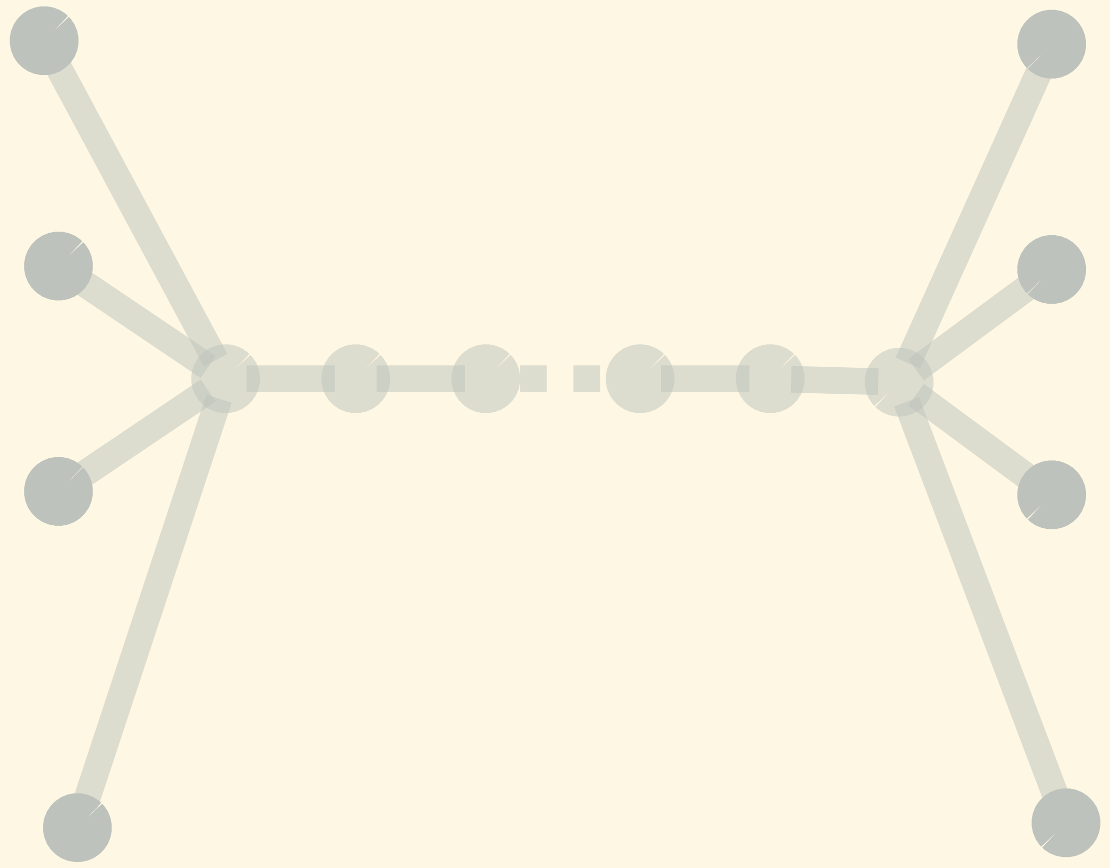


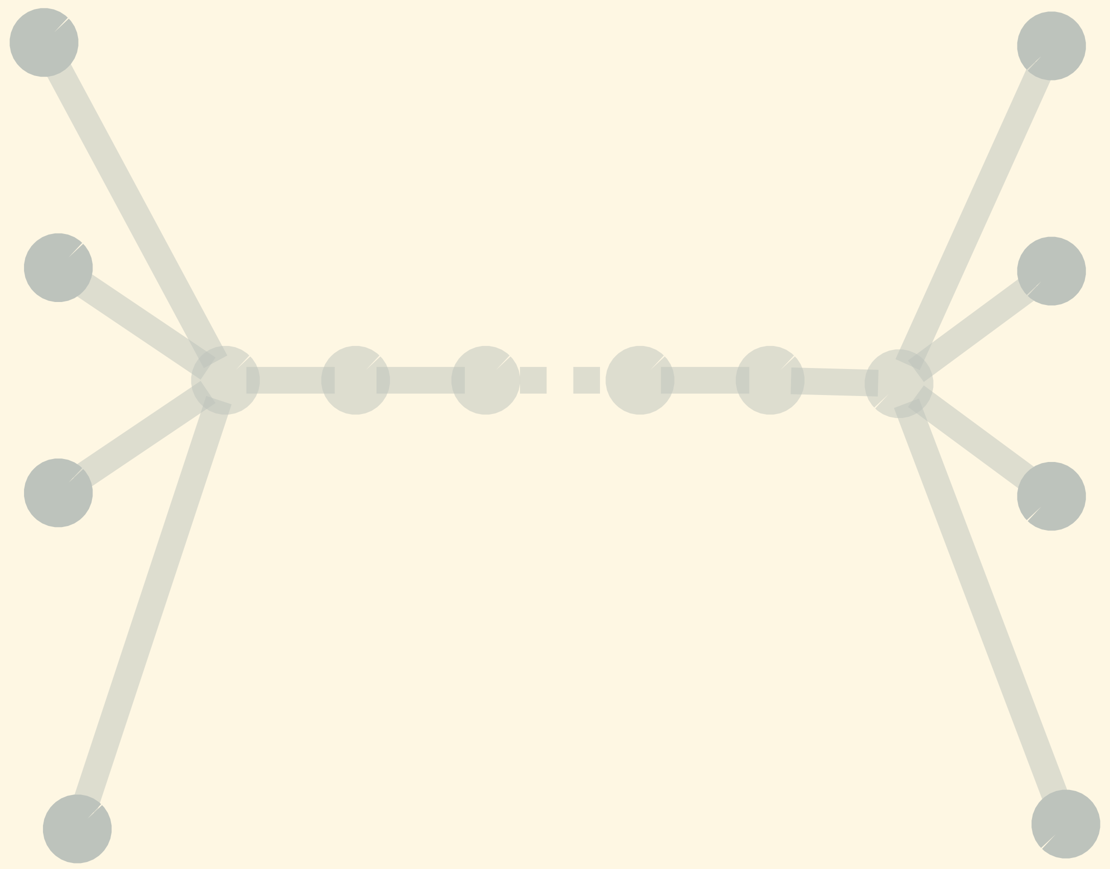


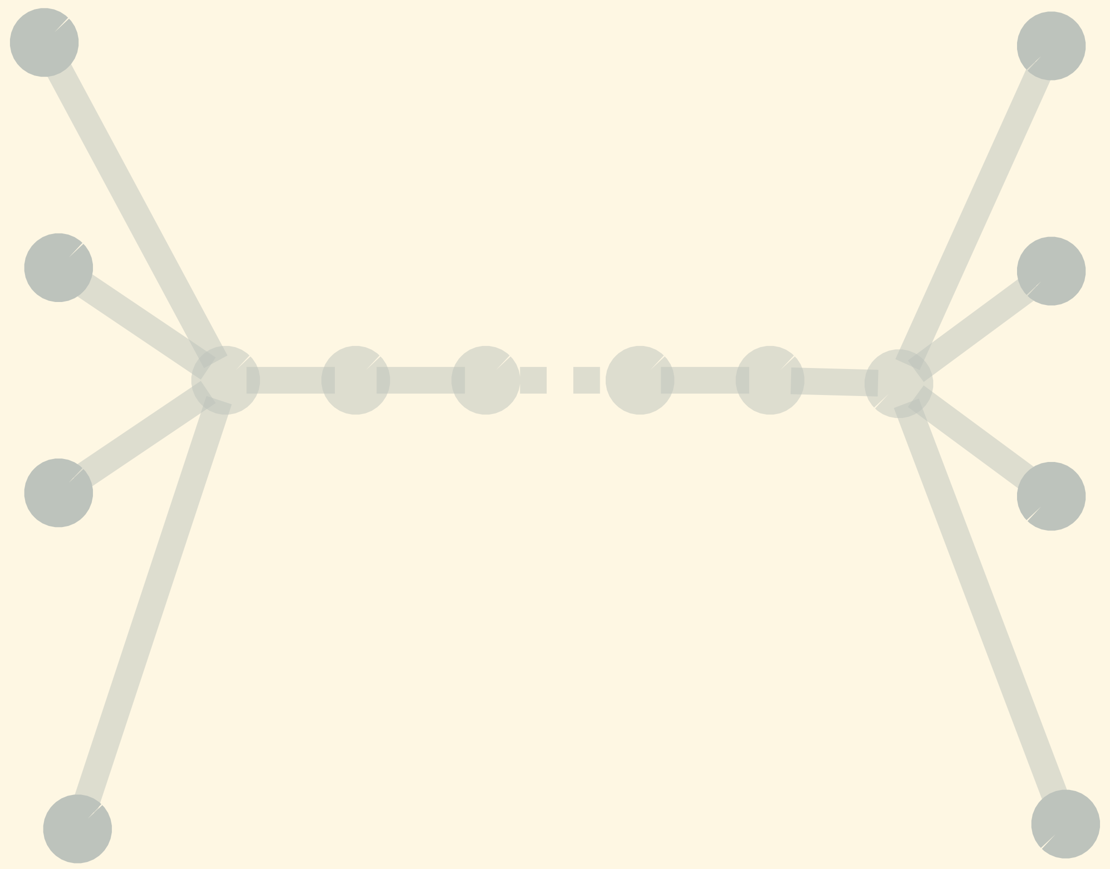


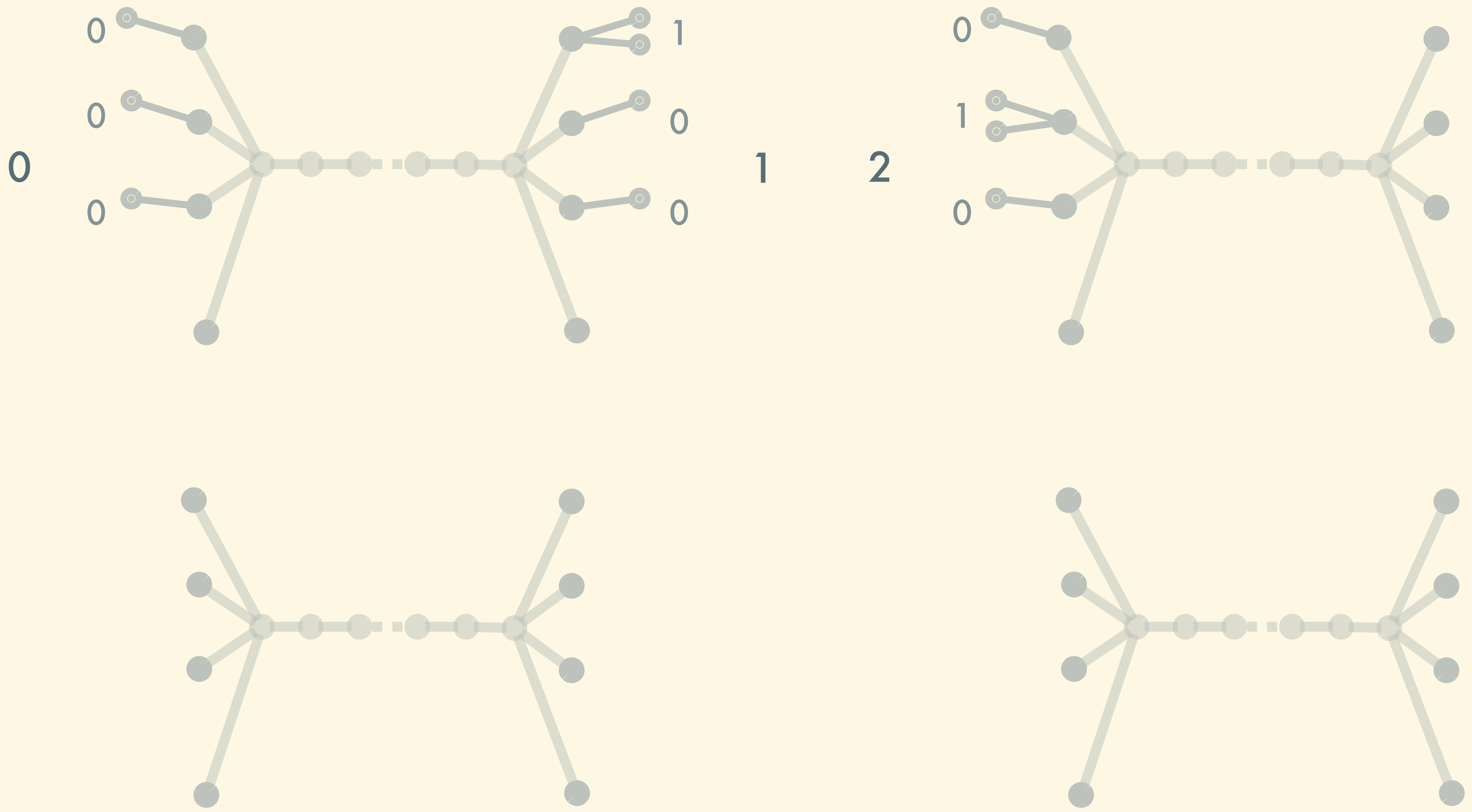

0
0
0
0

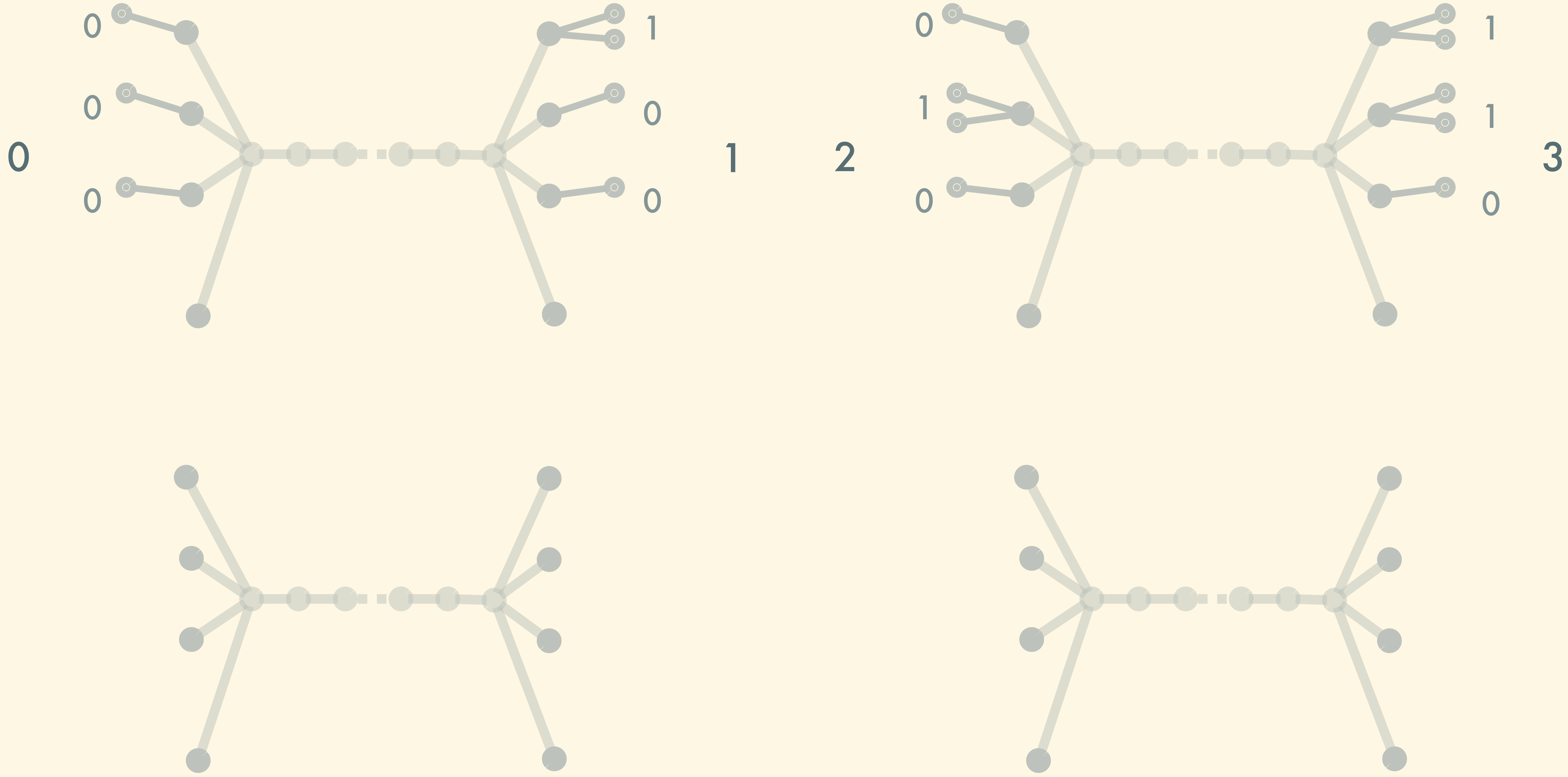

0
0
0
0
1
0
0
1



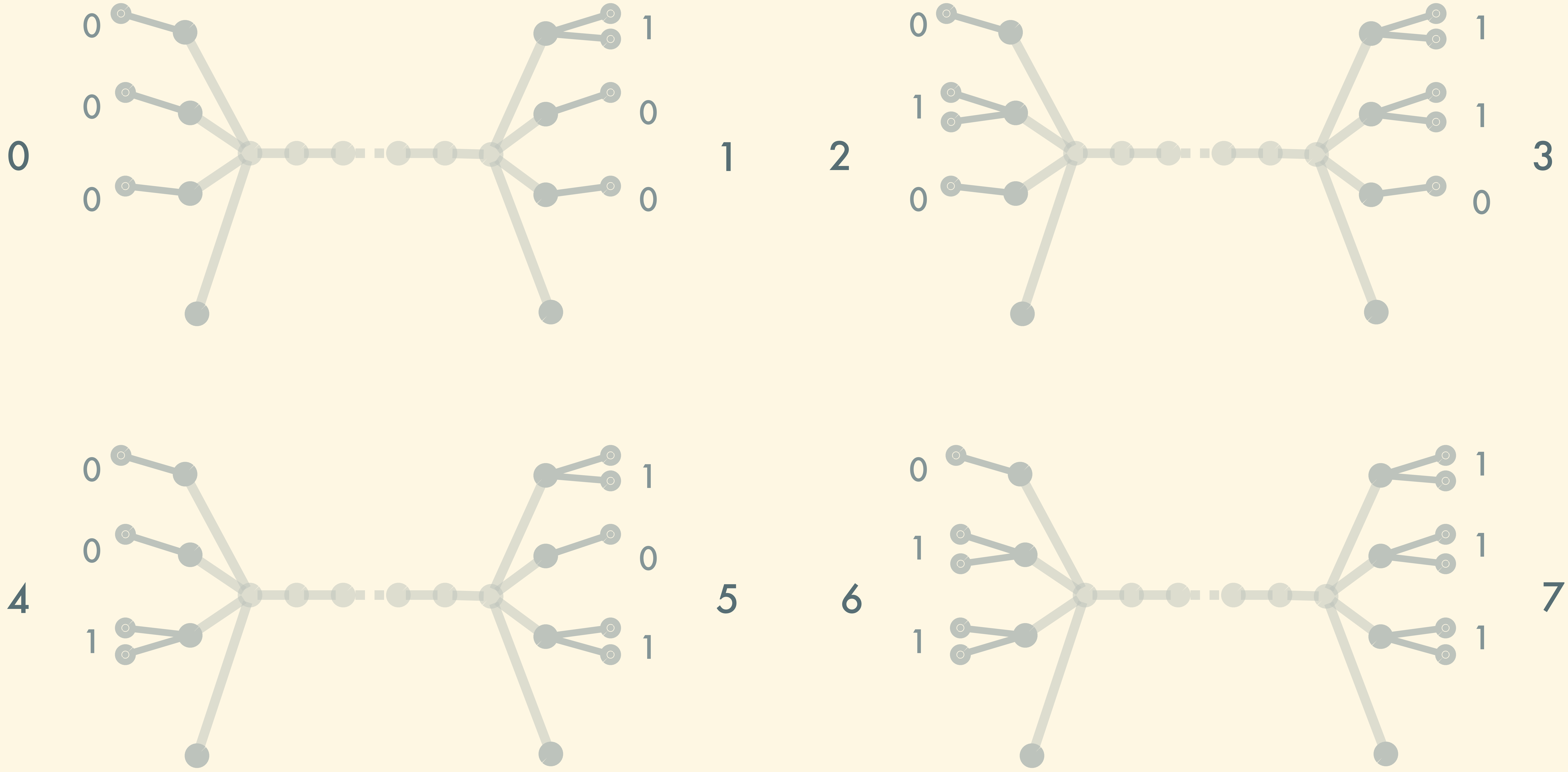




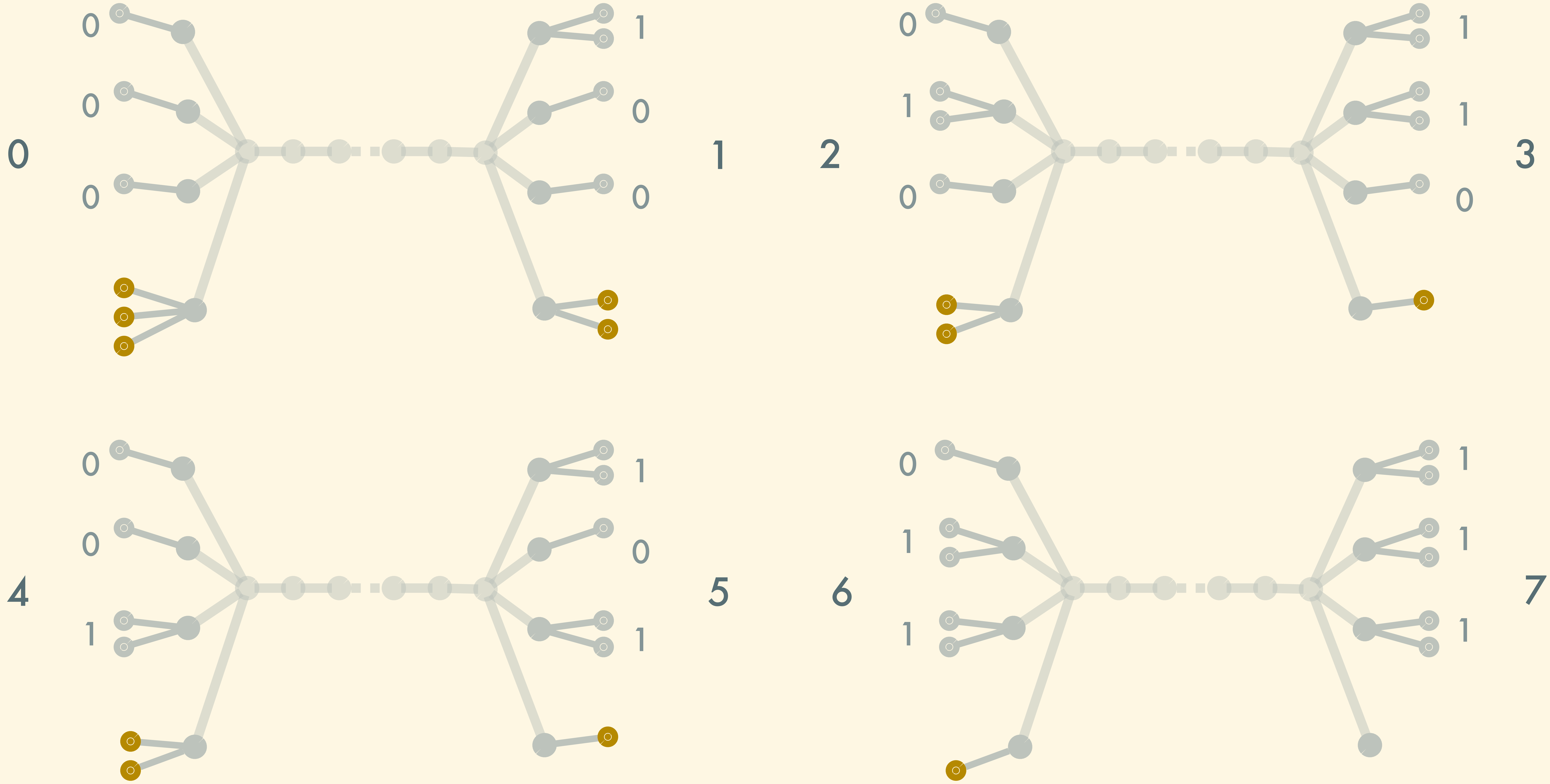

0
0
0
0
1
0
0
1
2
0
1
0



0
0
0
0
1
0
0
1
2
0
1
0
1
1
0
3



0
1
2
3
4
5
6
7


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

Double brooms

parameter δ

The number of leaves is determined by any bijective function f:

$f: \{1,\dots,Z\} \to \{1,\dots,\delta\}^{\delta}$

The number of trees in this family is δ^{δ}

REDUCTION

How to solve pair breaking problem using election in double brooms

- Hypothesis - It exists an algorithm ELECT that solves leader election:
  - on double brooms of parameter δ
  - in time D-2
  - using an oracle $\mathcal{O}$ that gives advices of size o(log log Z), $Z = \delta^\delta$
- Lets now describe:
  - The coloration function: using $\mathcal{O}$
  - The decision function used by Alice and Bob: using ELECT

Reduction

Coloration function C, the Oracle's job

Input integers: a b

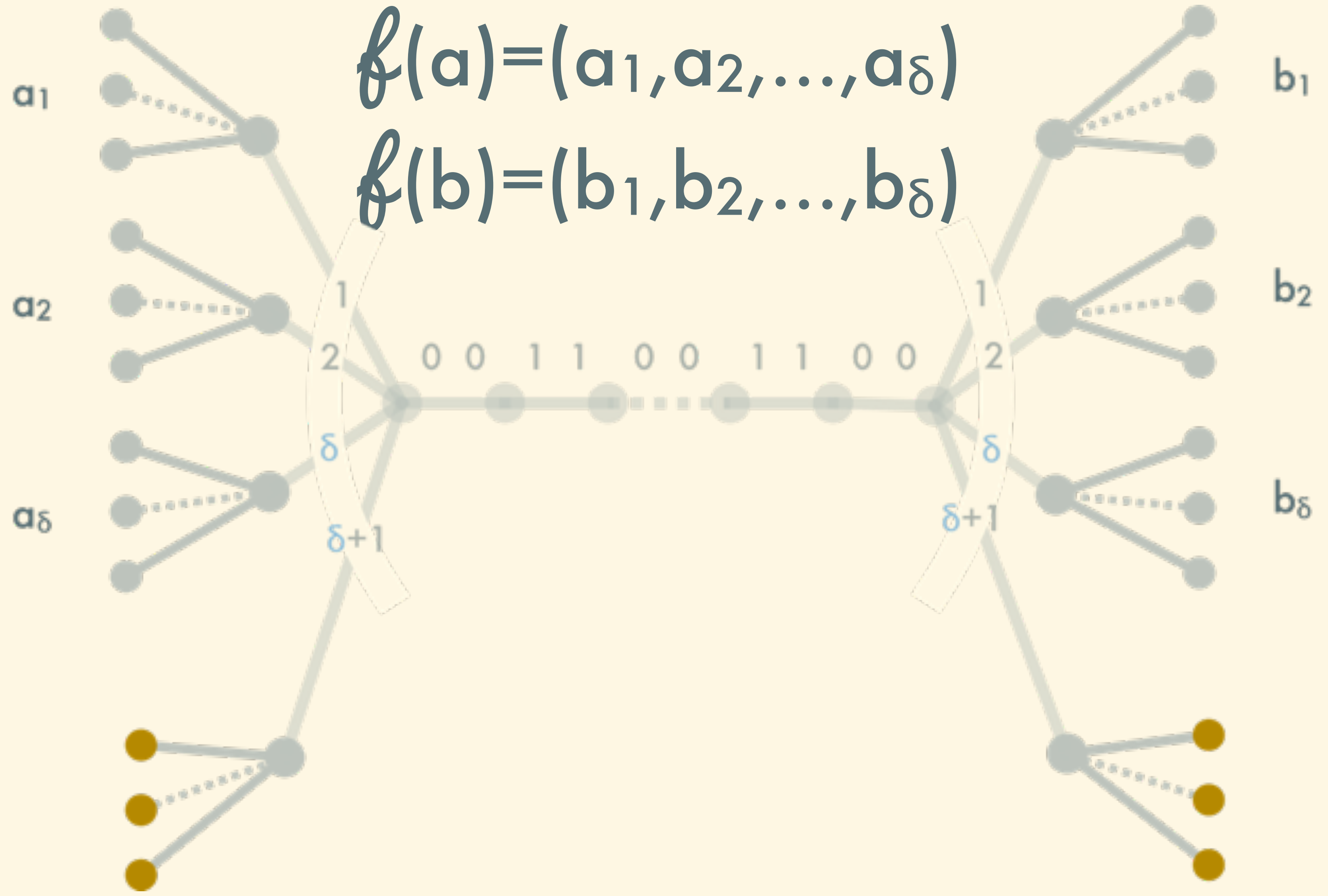
Output color: some advice

C:

Build the double broom for (a,b)

$f(a)=(a_1,a_2,\ldots,a_\delta)$

$f(b)=(b_1,b_2,\ldots,b_\delta)$



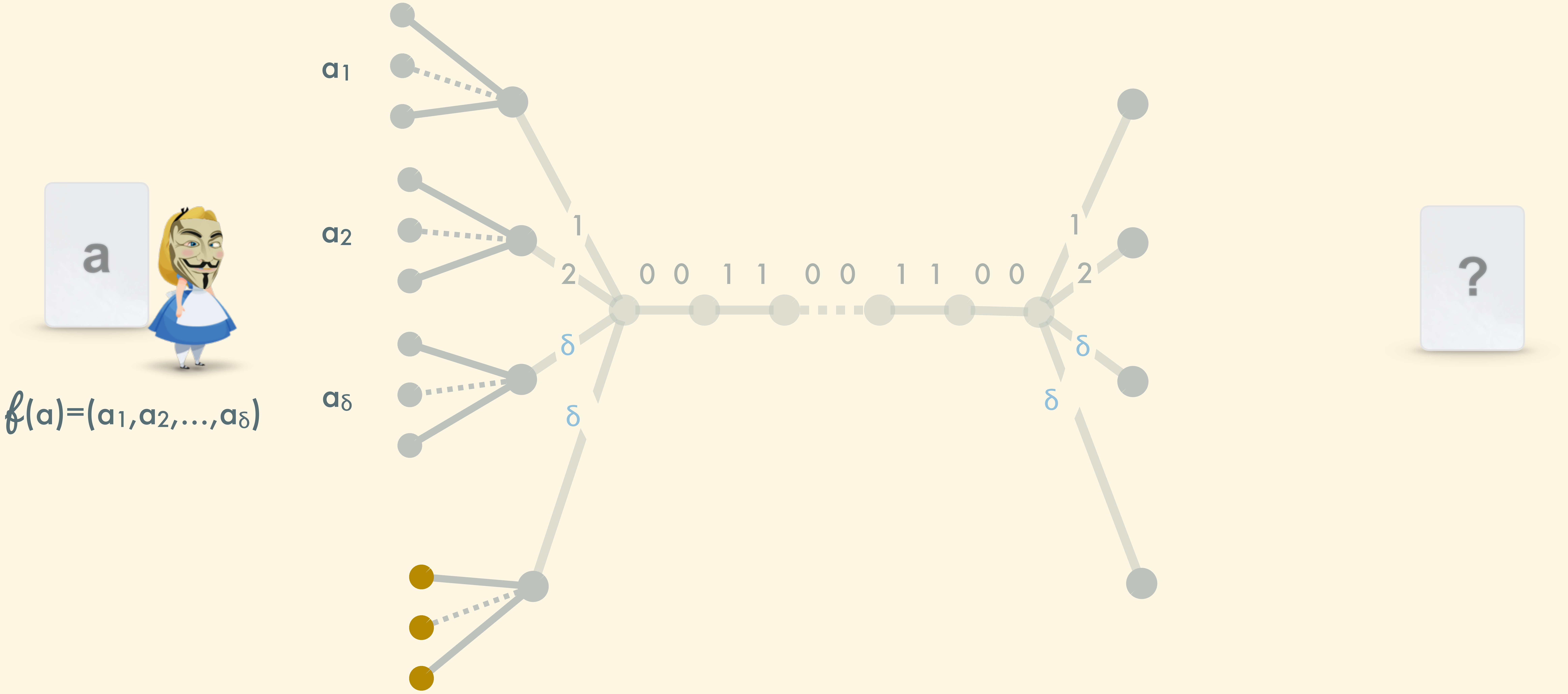
Give it to the oracle $\mathcal{O}$

Which output

some advice

Reduction

Players strategy - ~~Alice~~'s point of view



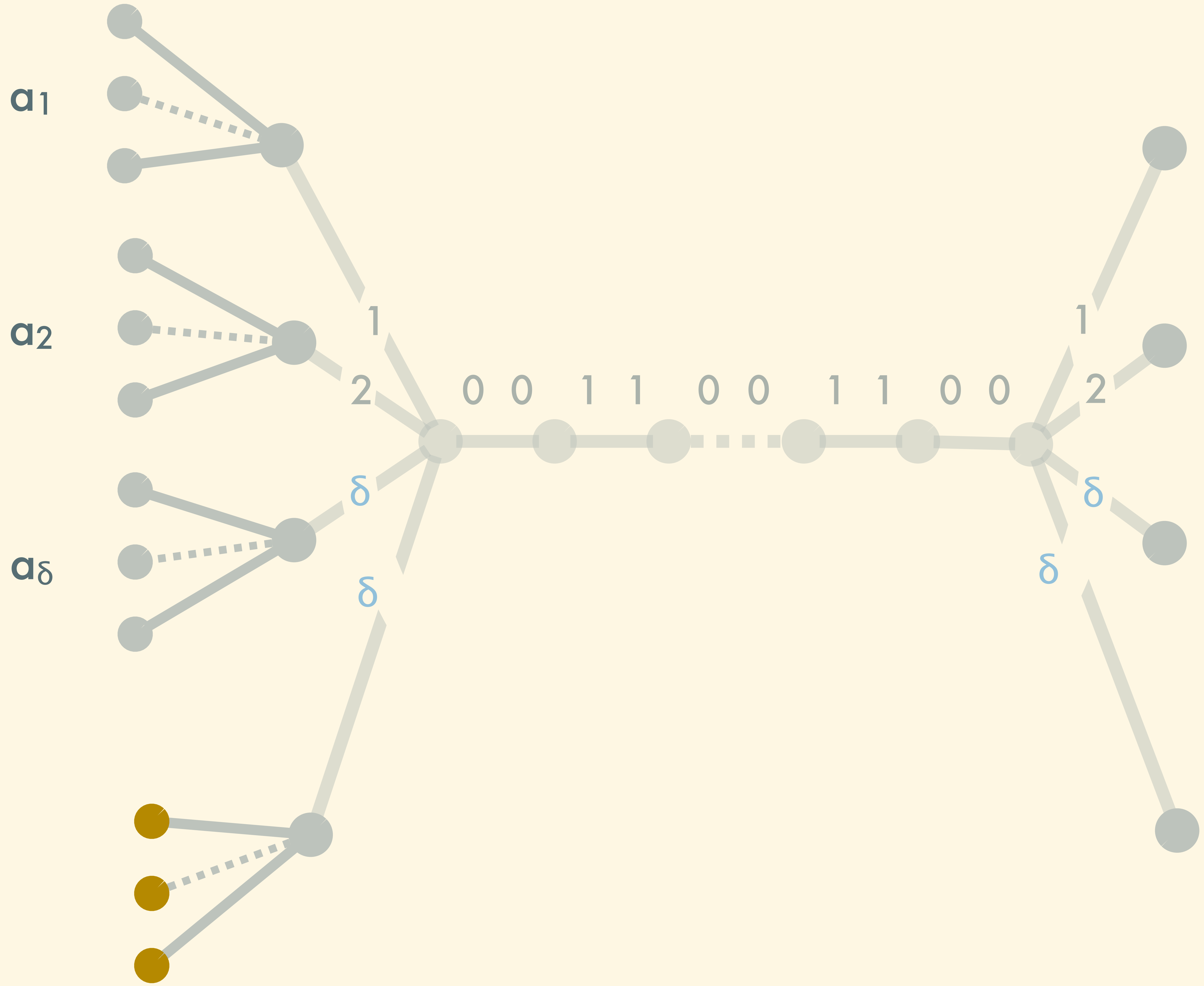
REDUCTION

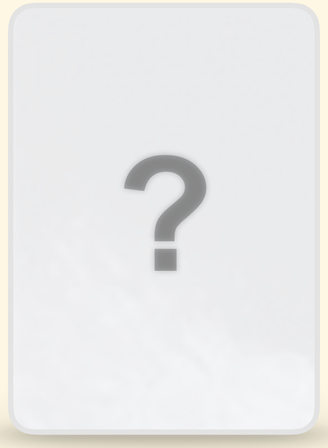
Players strategy - ~~Alice~~'s point of view

$f(a)=(a_1,a_2,\ldots,a_\delta)$

x = smallest integers different from a

$f(x)=(x_1,x_2,\ldots,x_\delta)$





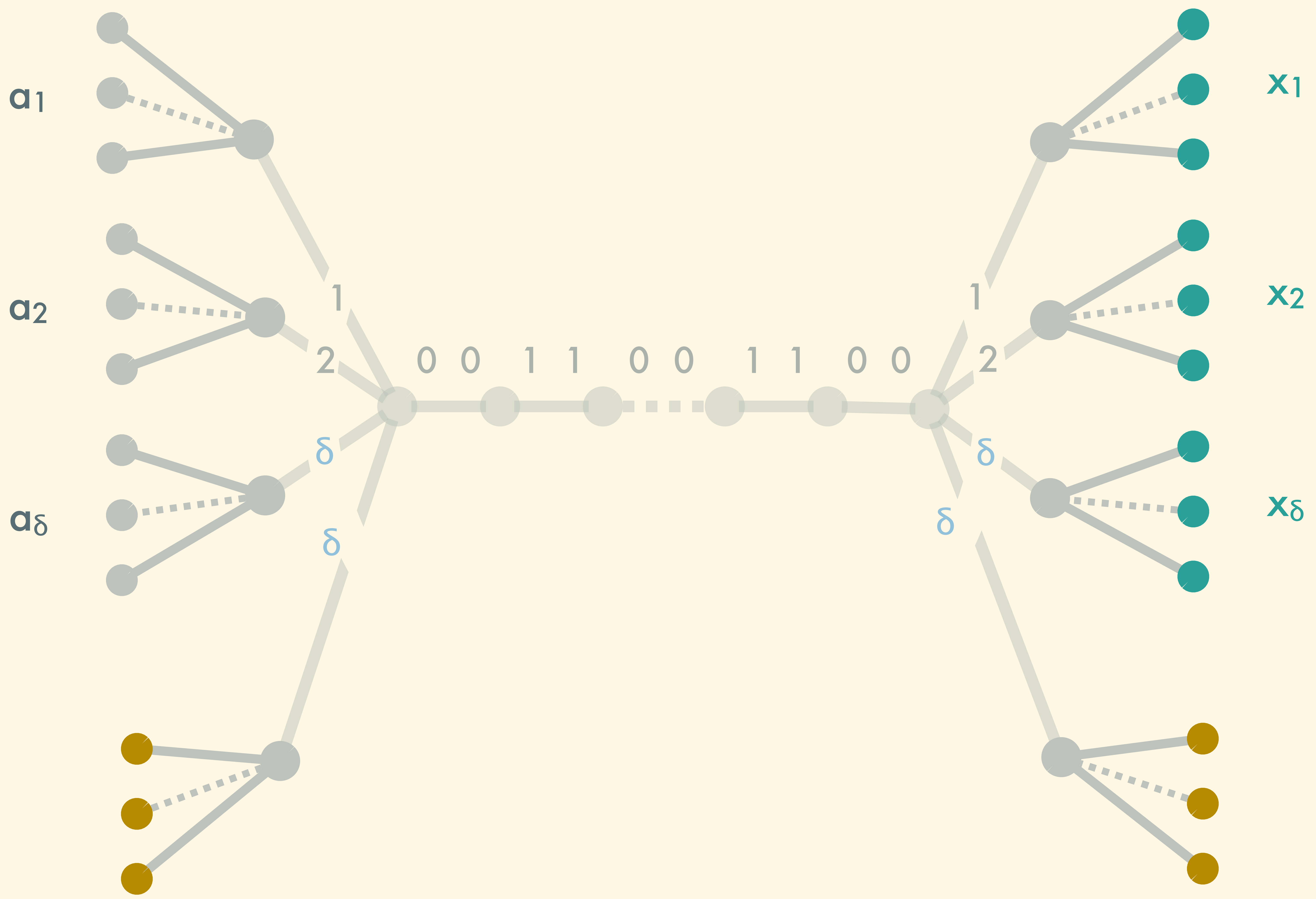
Reduction

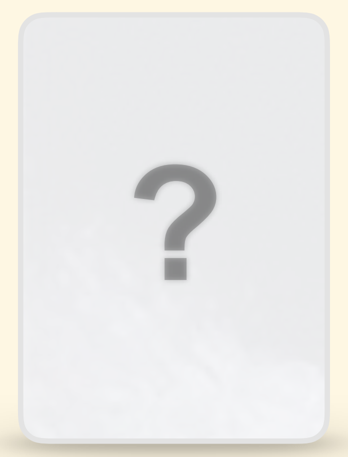
Players strategy - ~~Alice~~'s point of view

$f(a)=(a_1,a_2,\ldots,a_\delta)$

x = smallest integers different from a

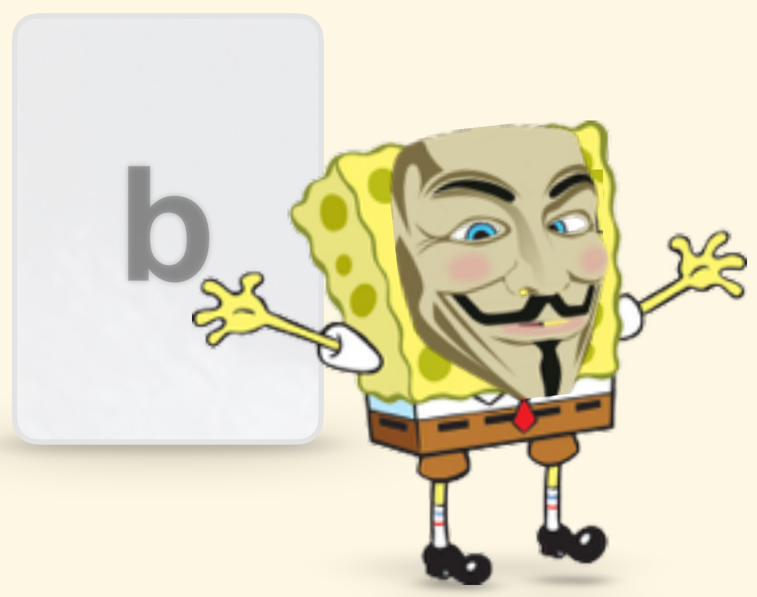
$f(x)=(x_1,x_2,\ldots,x_\delta)$



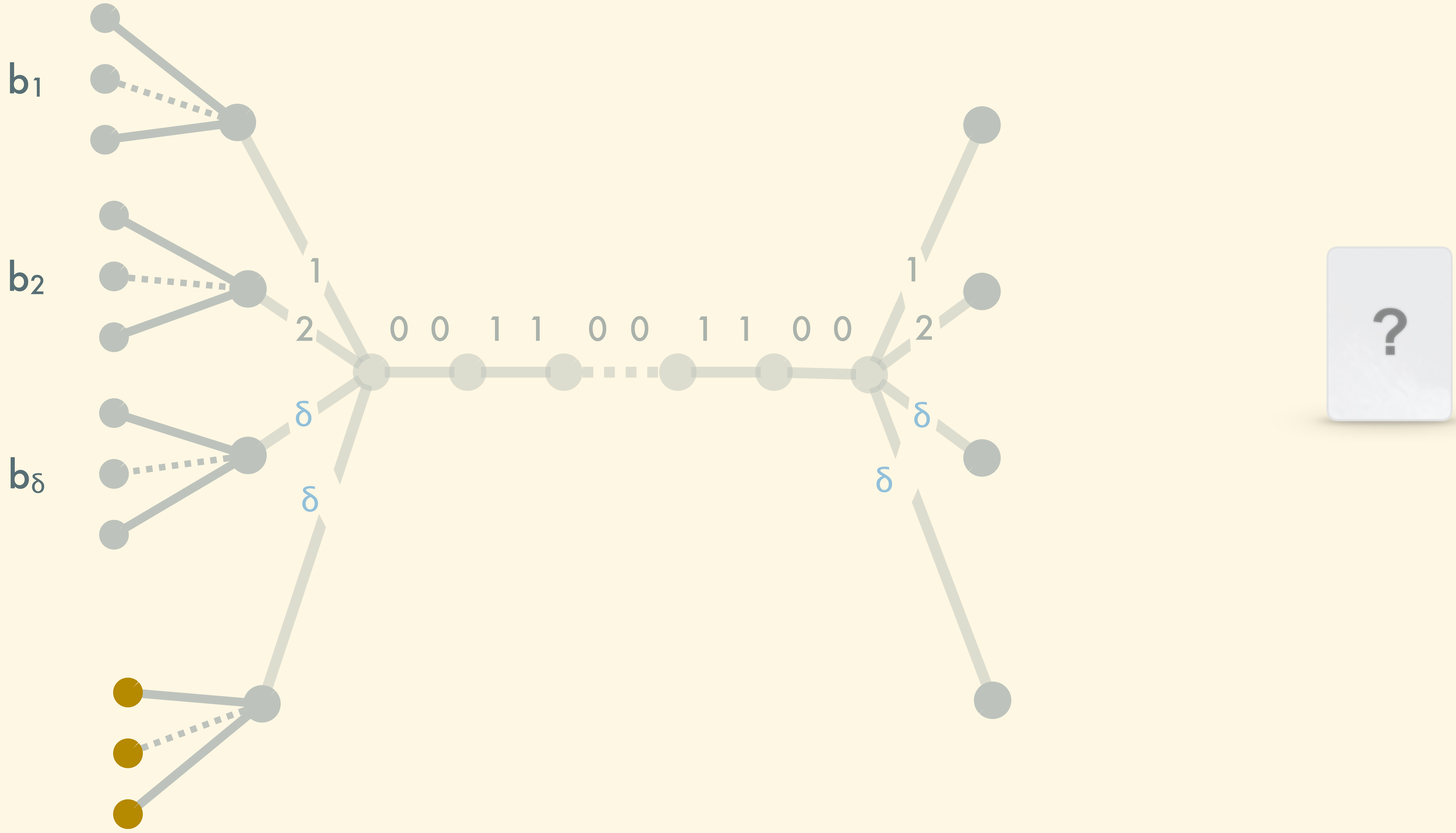


Reduction

Players strategy - ~~Bob~~'s point of view

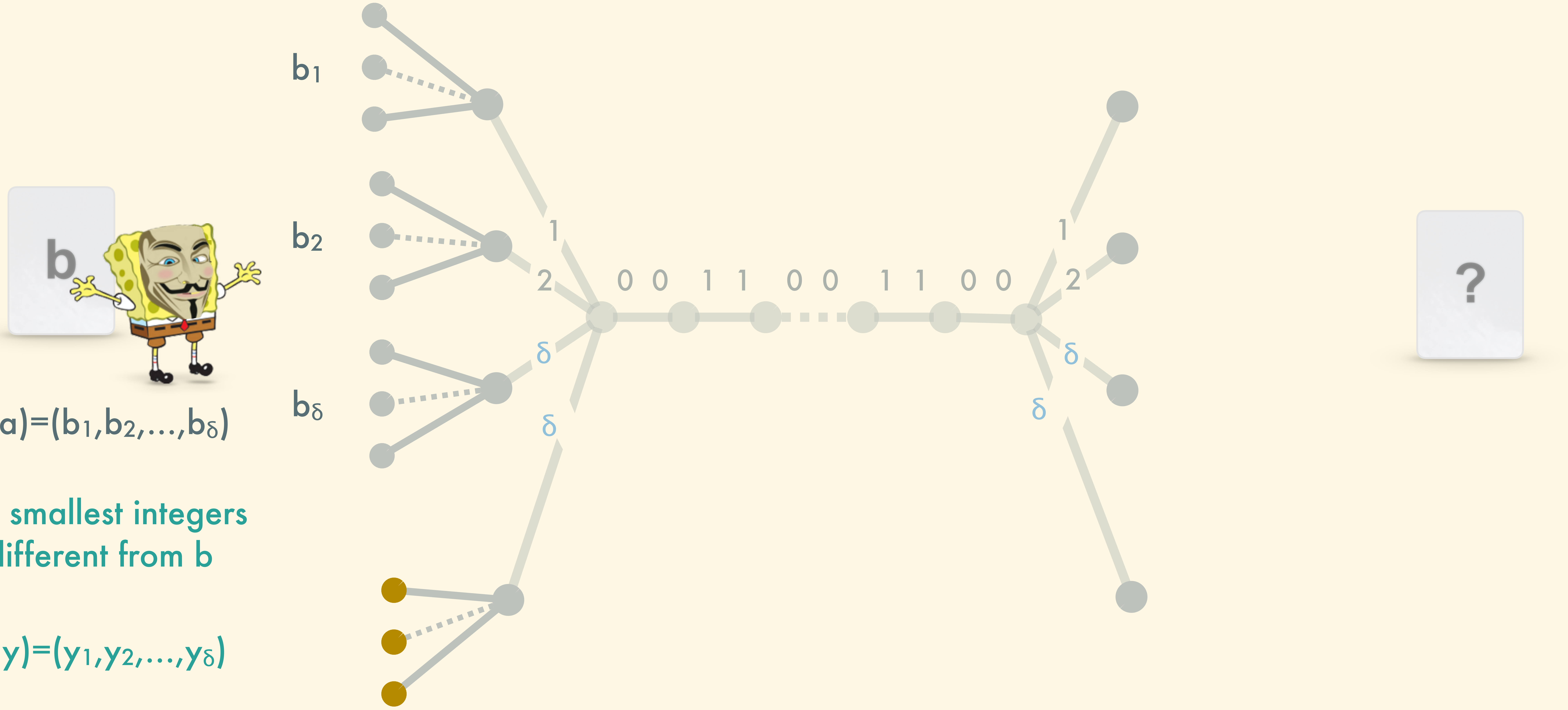


$f(a)=(b_1,b_2,\dots,b_\delta)$



Reduction

Players strategy - ~~Bob~~'s point of view



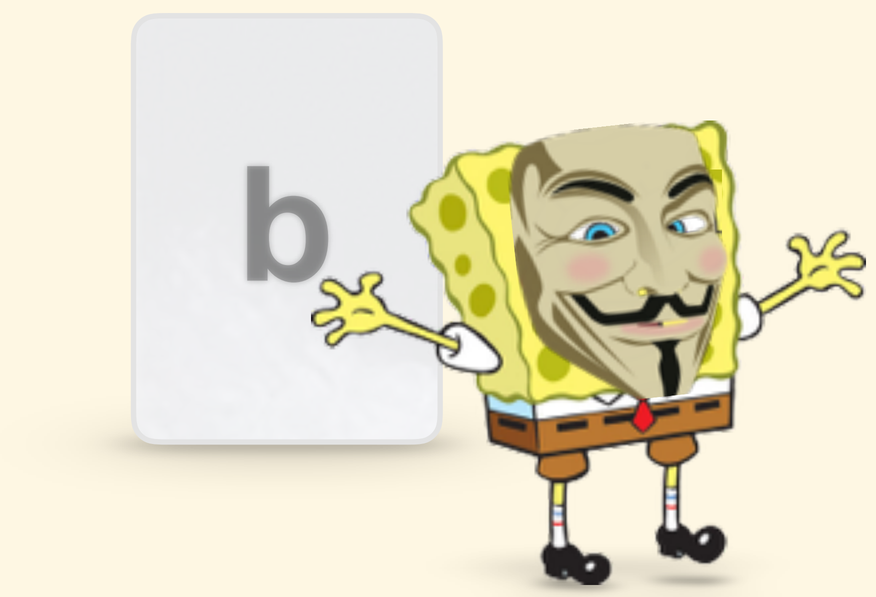
$f(a)=(b_1,b_2,\ldots,b_\delta)$

y = smallest integers different from b

$f(y)=(y_1,y_2,\ldots,y_\delta)$

REDUCTION

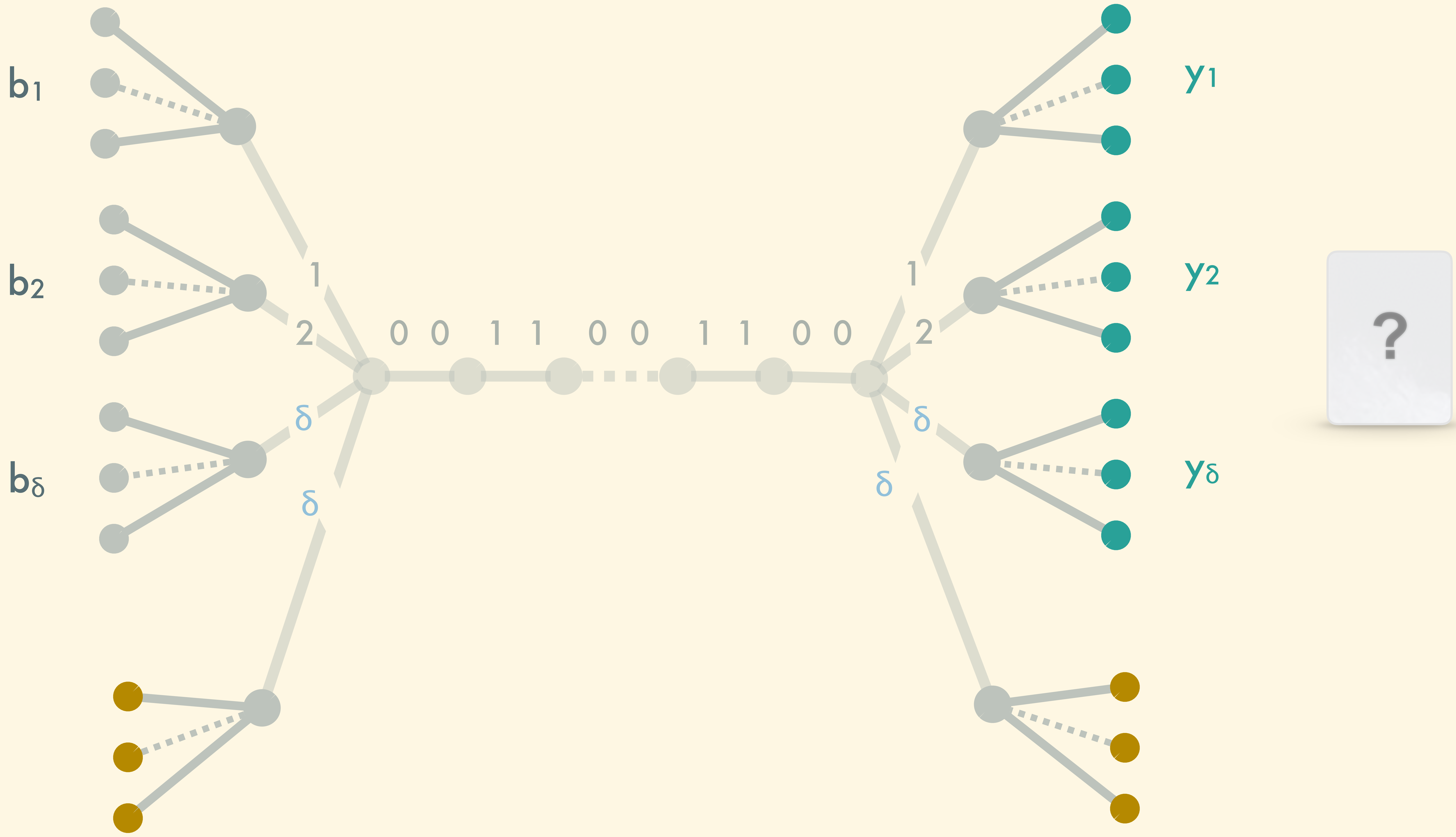
Players strategy - ~~Bob~~'s point of view



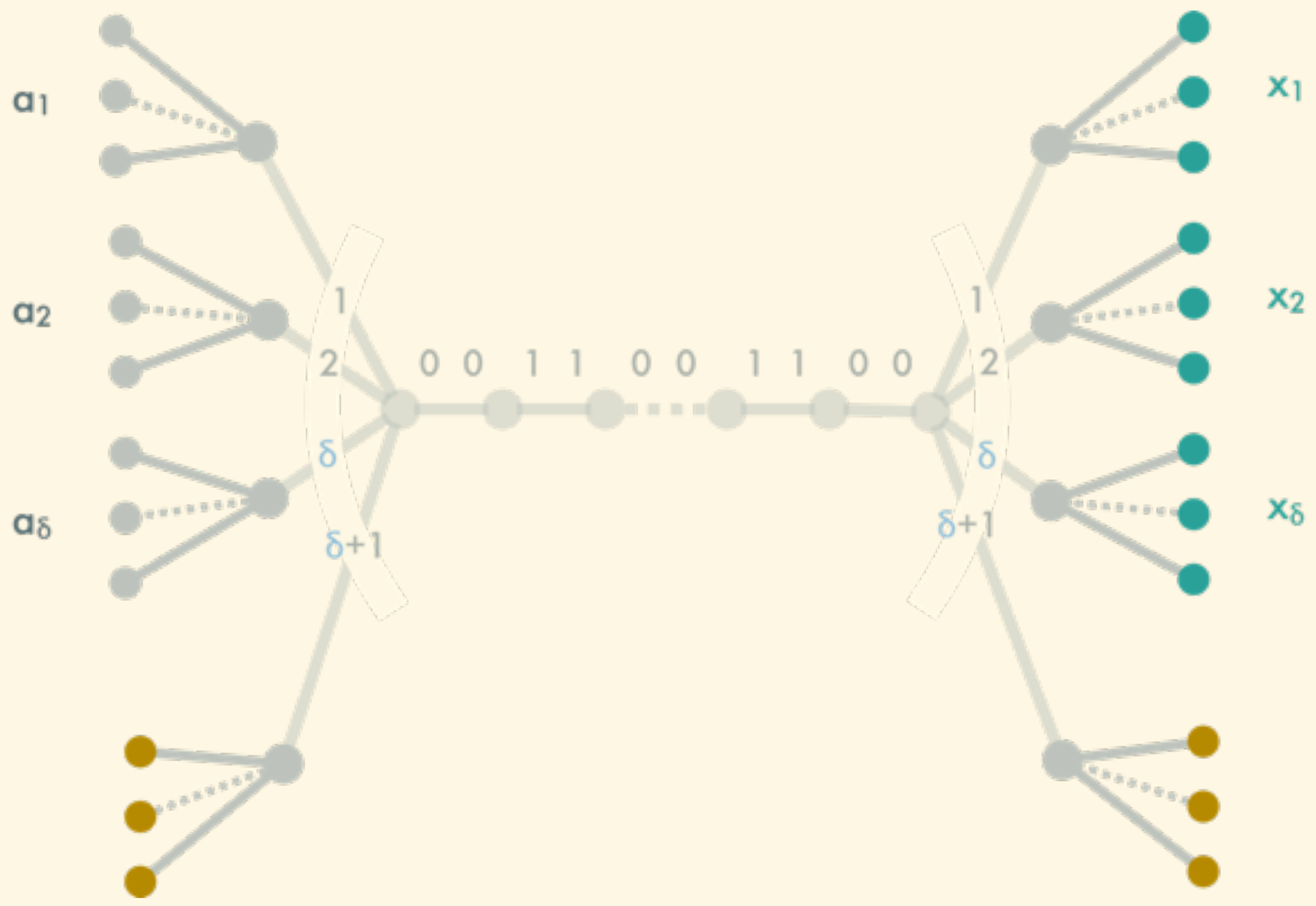
$\mathscr{f}(a)=(b_1,b_2,\dots,b_\delta)$

y = smallest integers different from b

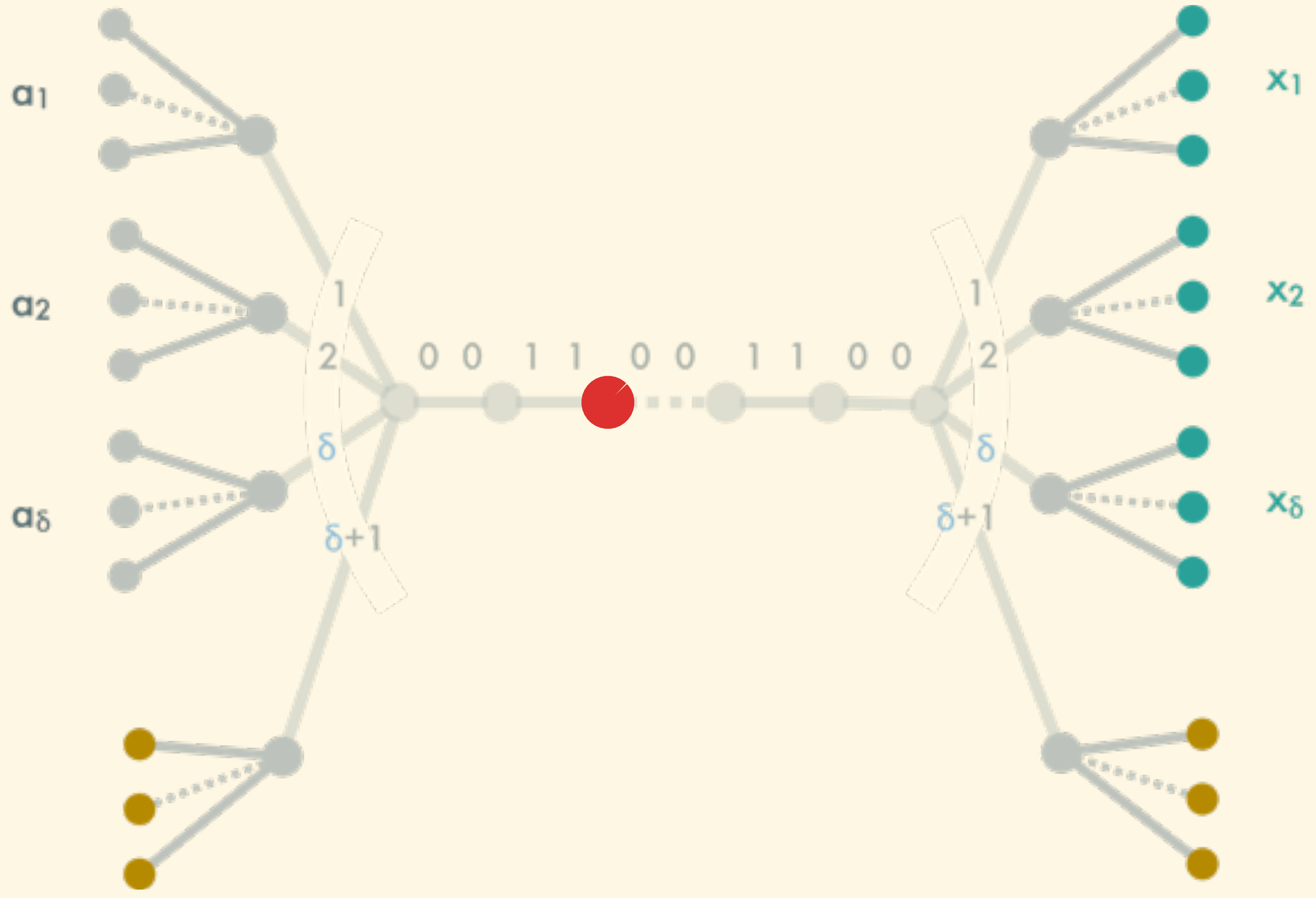
$\mathscr{f}(y)=(y_1,y_2,\dots,y_\delta)$



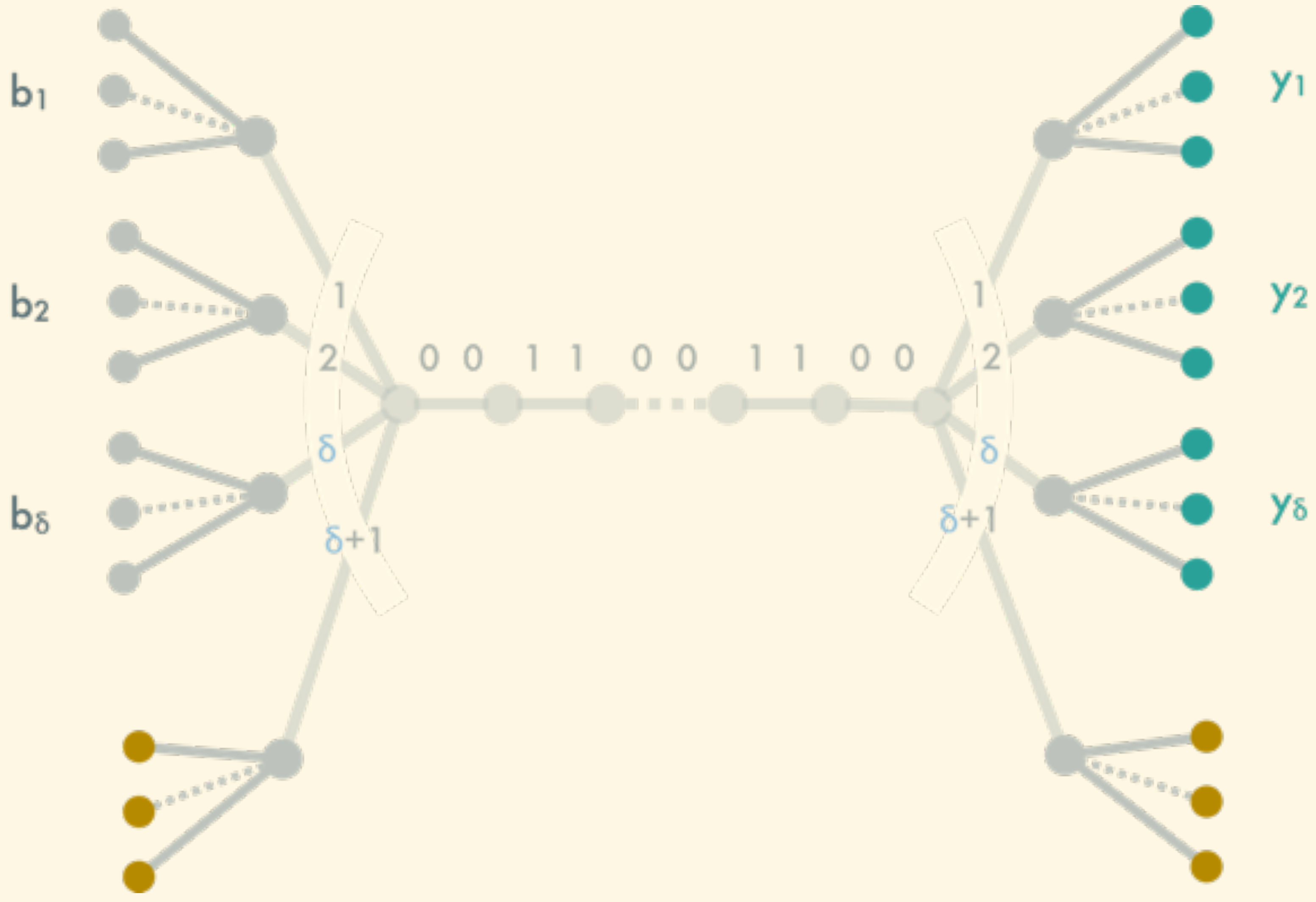
some advice +



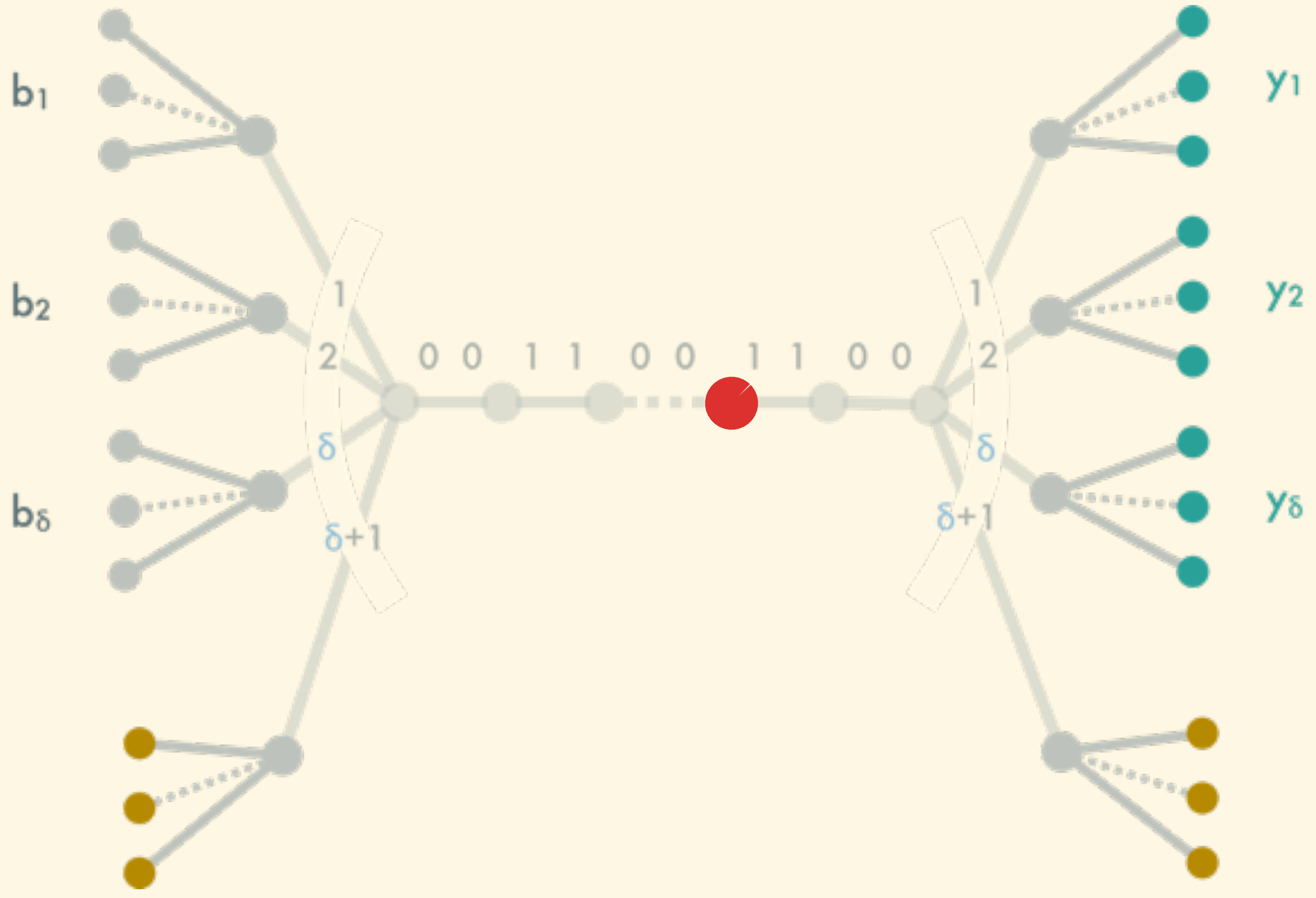
+ ELECT =



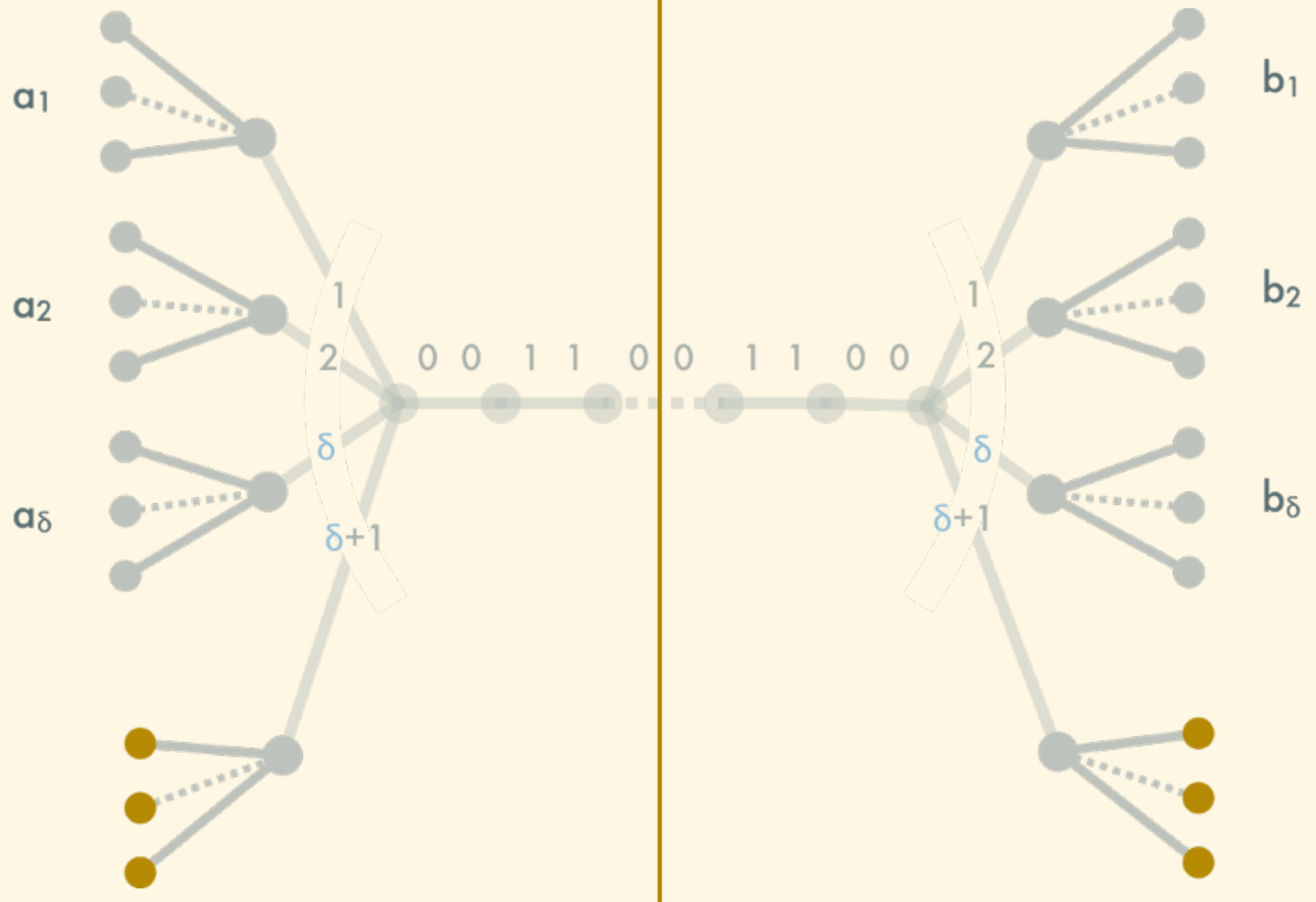
some advice +



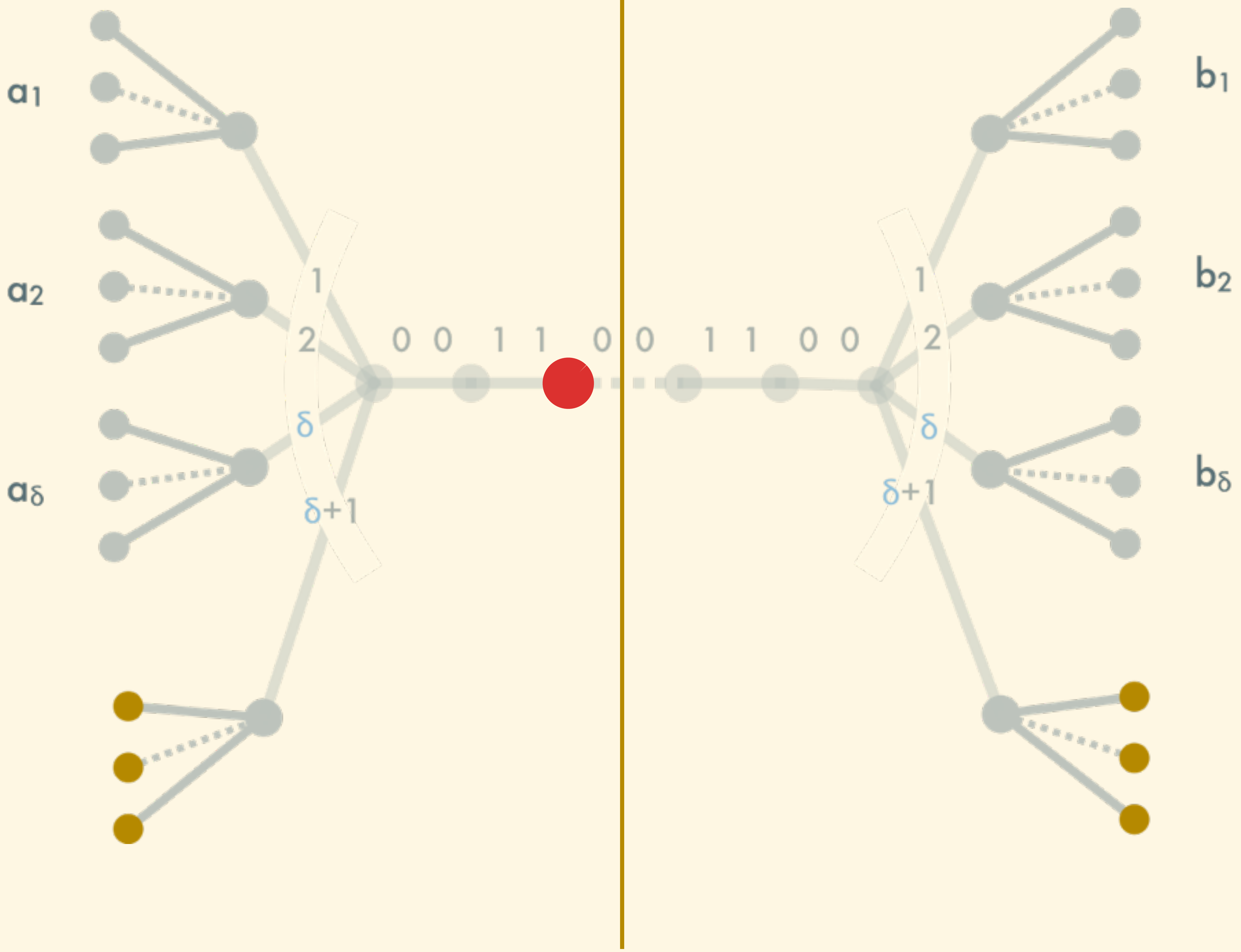
+ ELECT =



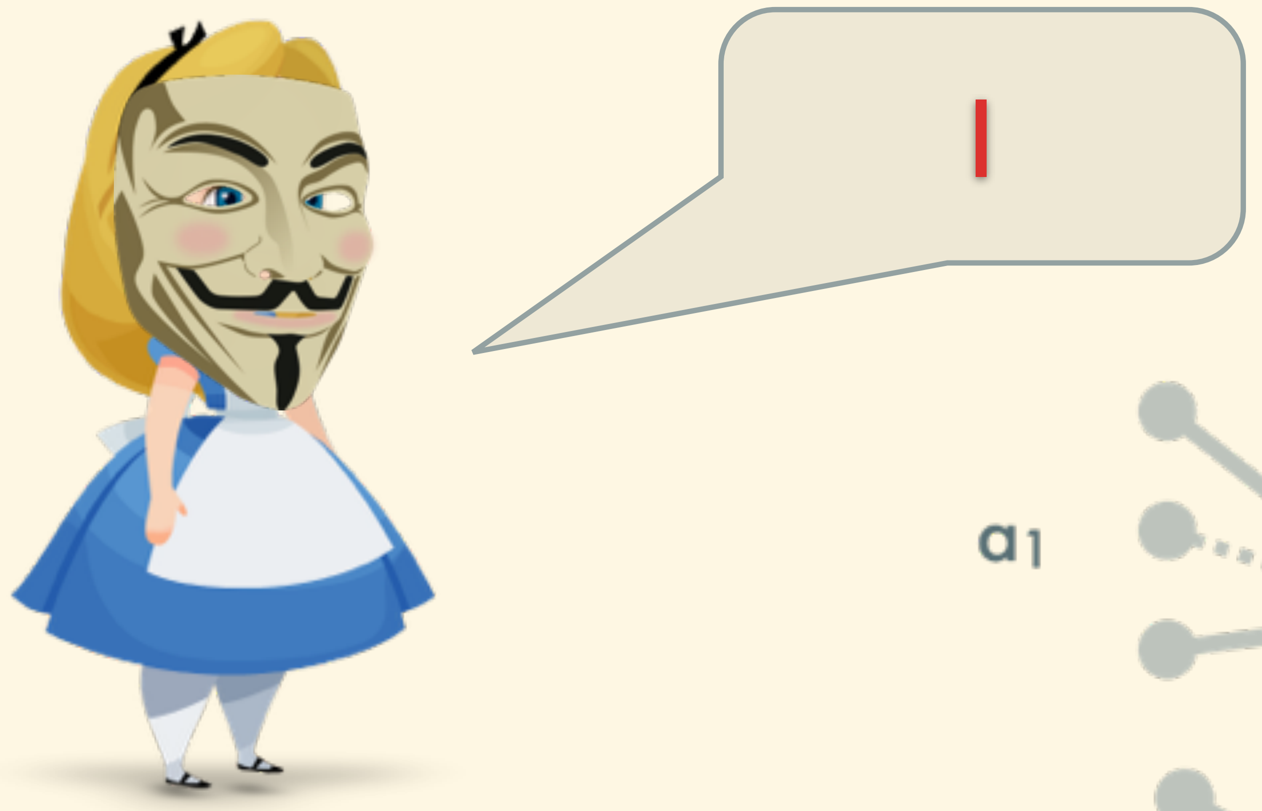
Breaking the symmetry

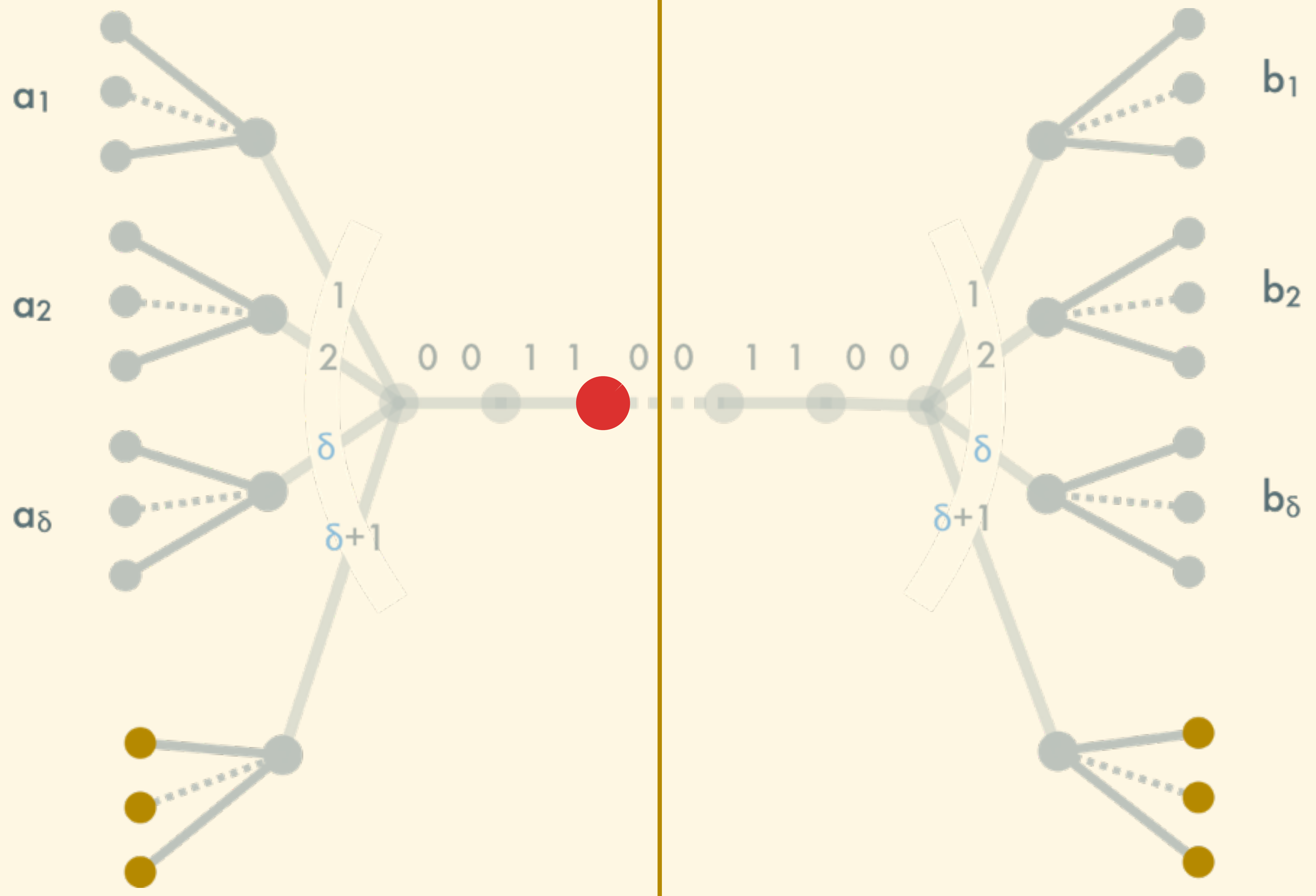


Breaking the symmetry

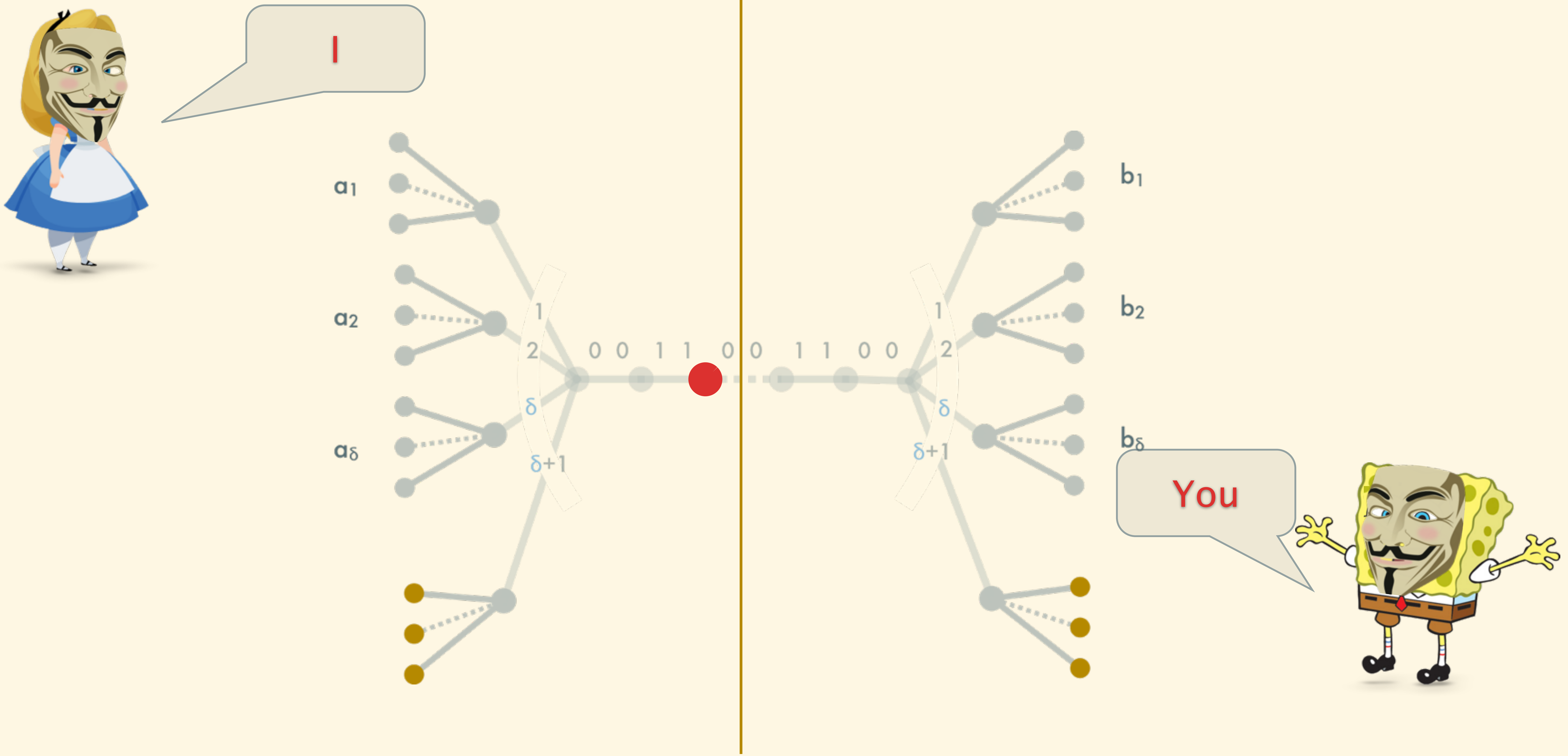


Breaking the symmetry





Breaking the symmetry



Lower bound conclusion

- We solved the pair breaking problem using only o(log log Z) bits, with $Z = \delta^{\delta}$, a contradiction, therefore the assumption that ELECT exists is false.
- Make $\delta = \sqrt{n} \Rightarrow$ size of double brooms is n,
- and the bound on the size of the advice turns out to be:
  - $\Omega(\log\log \delta^{\delta}) \in \Omega(\log\log \sqrt{n}^{\sqrt{n}}) \in \Omega(\log n)$

RESULTS

Time	Advice size	Remark
D	0	
D-1	$\Theta(\log D)$	
D-2	$\Theta(\log D)$ $\Theta(\log n)$	D even D odd
[β*D, D-3], β > 1/2	$O(n \log n/D)$ $\Omega(n/D)$	upper bound D even or τ < D-3
α*D, α < 1/2	$\Theta(n)$	D not too small ($D \in \omega(\log^2 n)$)

Open problems

- Lower bound for τ = D-3 when D is odd
- Lower and upper bounds for times close to D, $\tau = D \pm o(D)$
- Lower and upper bounds for small diameters $D \in O(\log^2 n)$
- Lower bound for $\tau \in o(\mathrm{diam})$.
  - At first sight it may seems like the bound $\Omega(n)$ still holds, but it's not the case because it is difficult to build a large family of trees in which election is still possible.

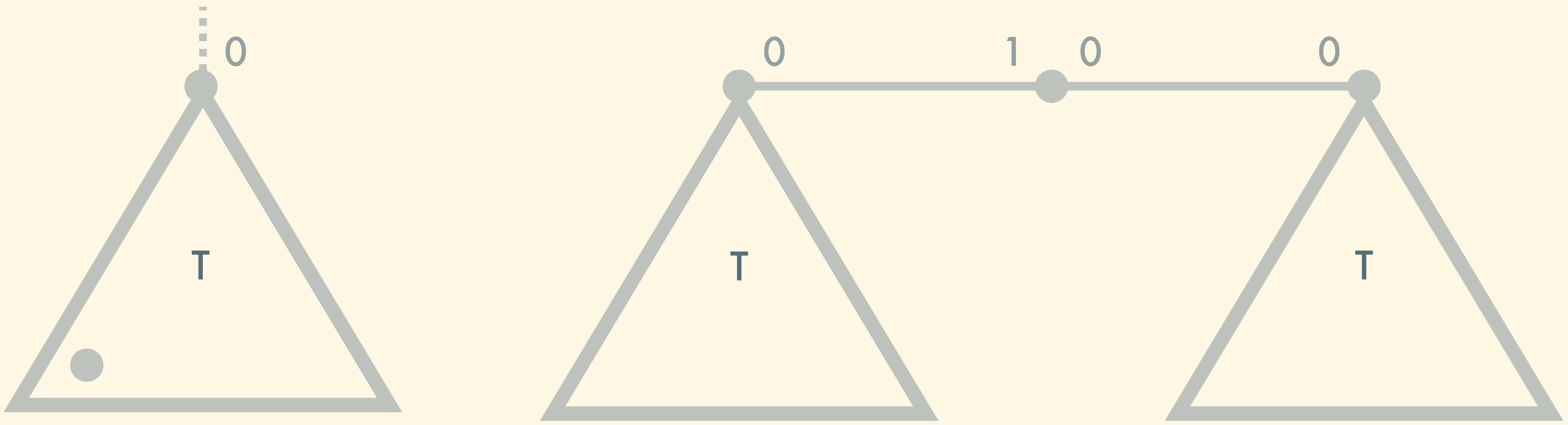
MERCI

Questions ?

QUESTION N° 1

Can you show a tree where election is not possible in time τ ?

An example for $\tau < \mathrm{D}/2$



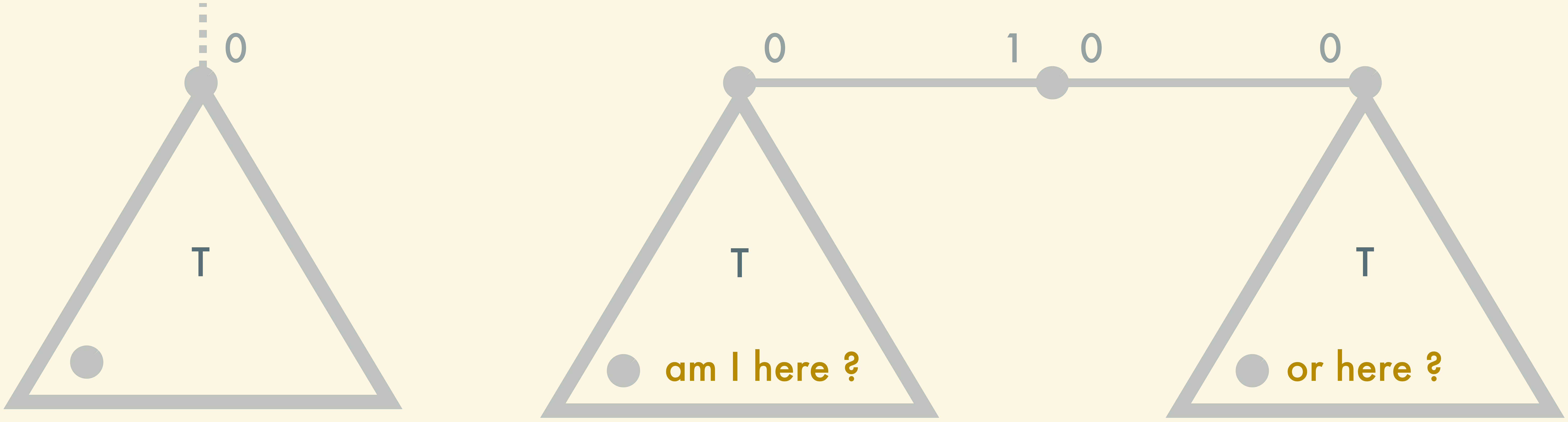
Node local knowledge

Oracle's advice (whole graph)

Question n° 1

Can you show a tree where election is not possible in time τ ?

An example for $\tau < D/2$



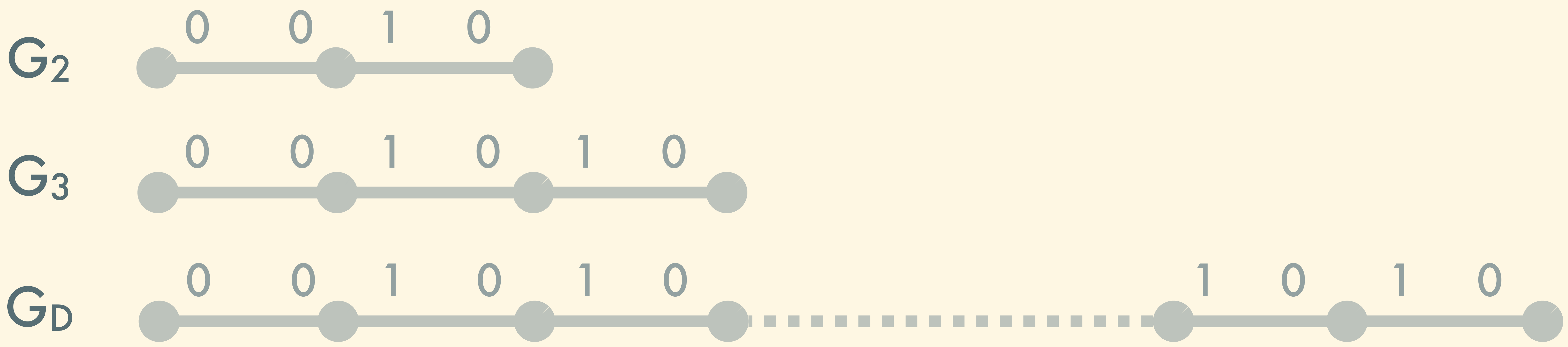
Node local knowledge

Oracle's advice (whole graph)

Question n°2

Does the algorithm has to know D ? Is it given fro free, etc. ?

Path-graphs family of size D-1, election in time in G_i is i-1:



Suppose we have < D-1 different advices, then two graphs have the same advice: