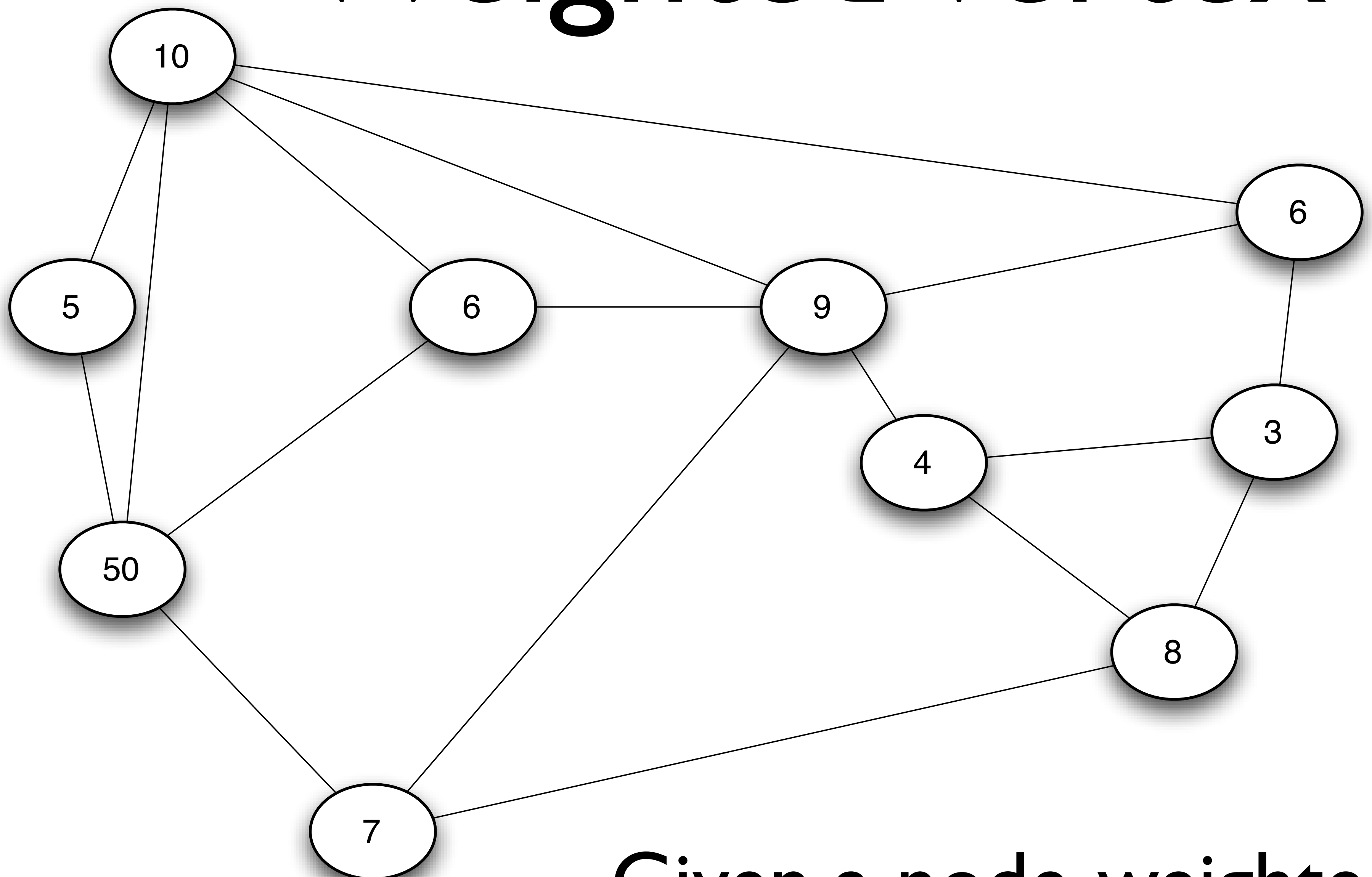
Distributed 2-approximation algorithm for Vertex Cover

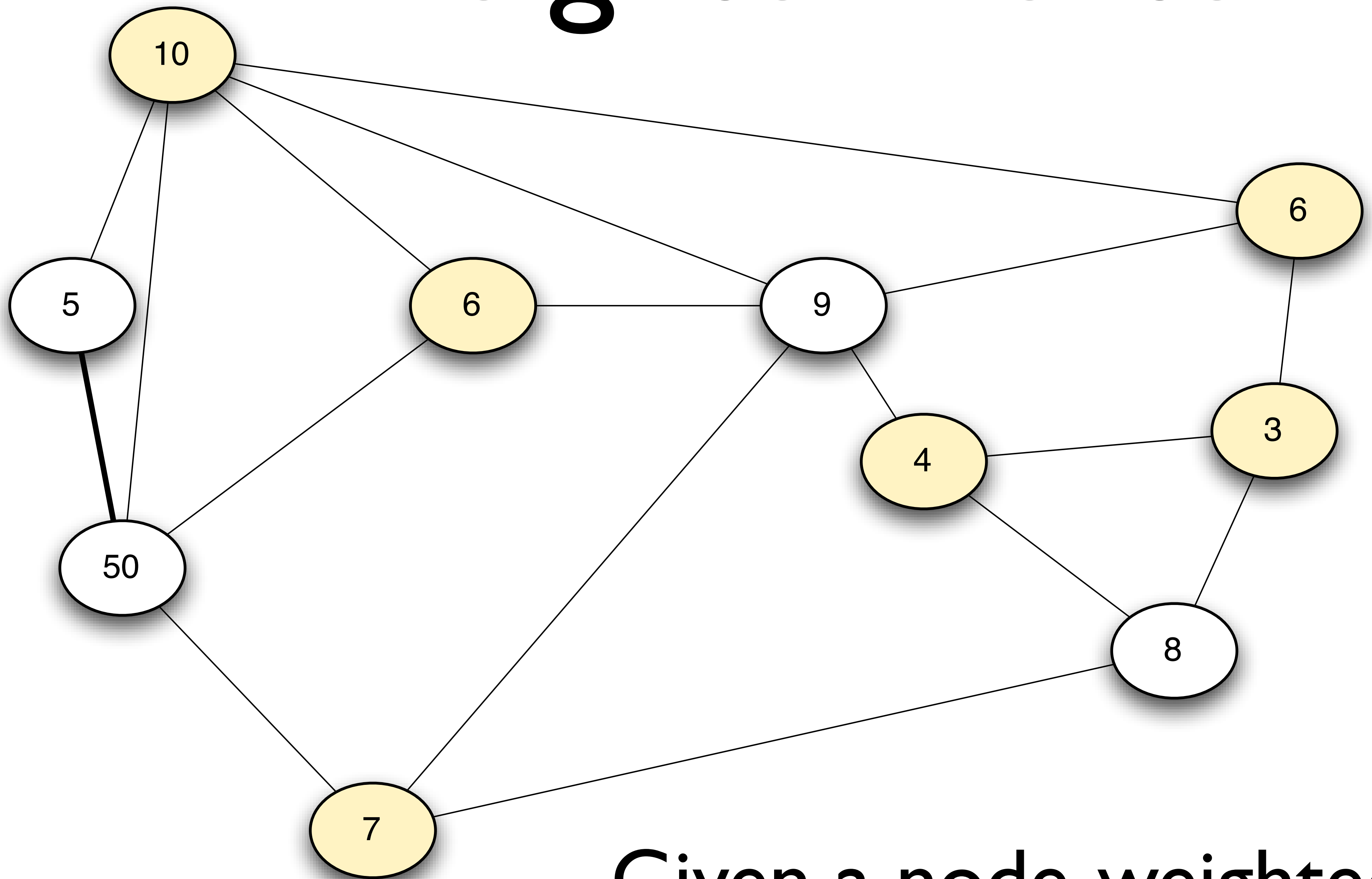
Christos Koufogiannakis and Neal E. Young
University of California, Riverside

Weighted Vertex Cover



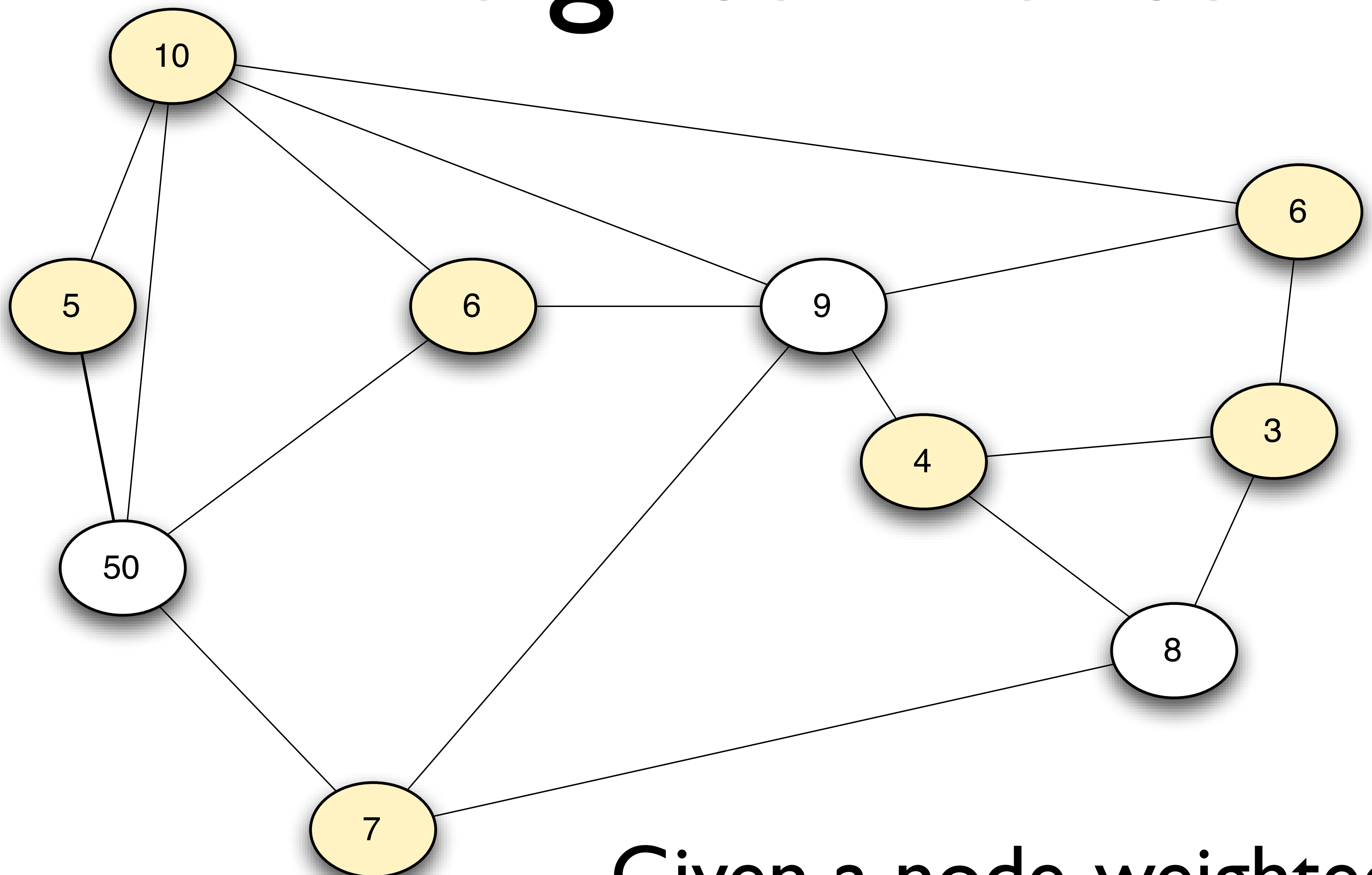
Given a node-weighted graph, find a minimum-weight subset of the nodes, touching all the edges.

Weighted Vertex Cover



Given a node-weighted graph, find a minimum-weight subset of the nodes, touching all the edges.

Weighted Vertex Cover



Given a node-weighted graph, find a minimum-weight subset of the nodes, touching all the edges.

A sequential 2-approximation algorithm

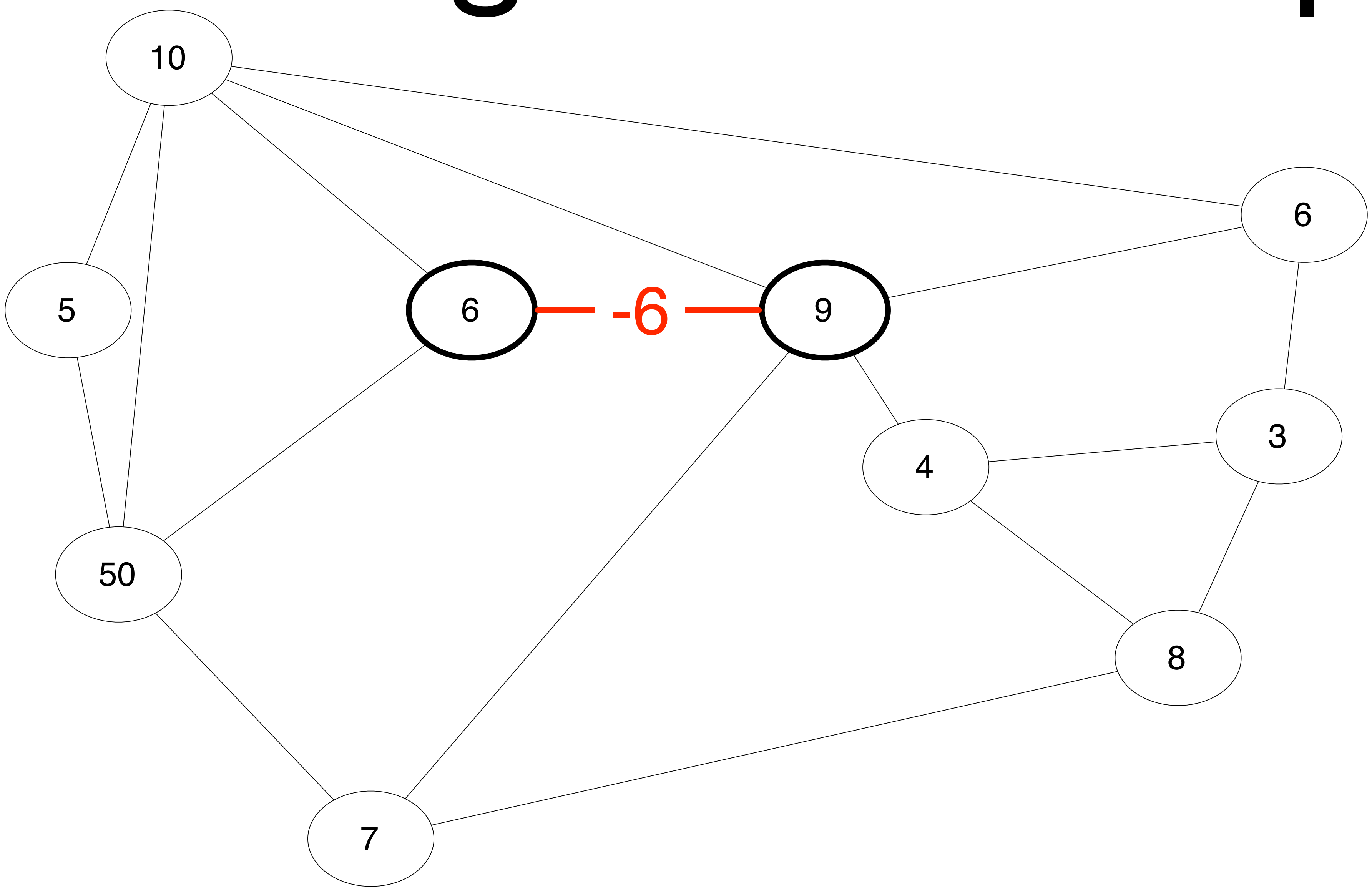
- “Edge discount” ---

  reduce edge endpoints’ costs equally.

- Do edge discounts until zero-cost nodes form a cover. Return the cover formed by the zero-cost nodes.

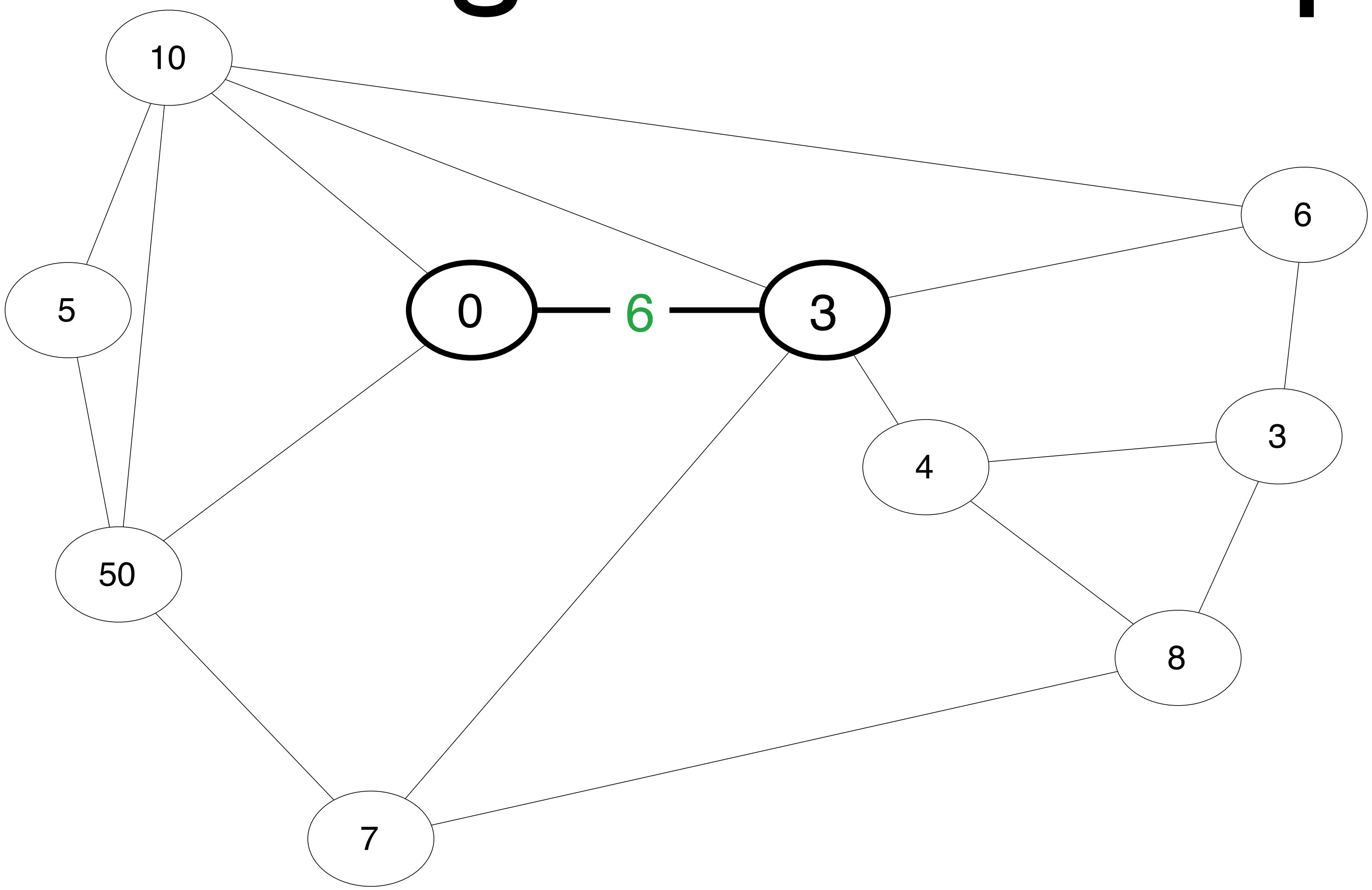
[Bar-Yehuda and Even, 1981]

edge discount operation



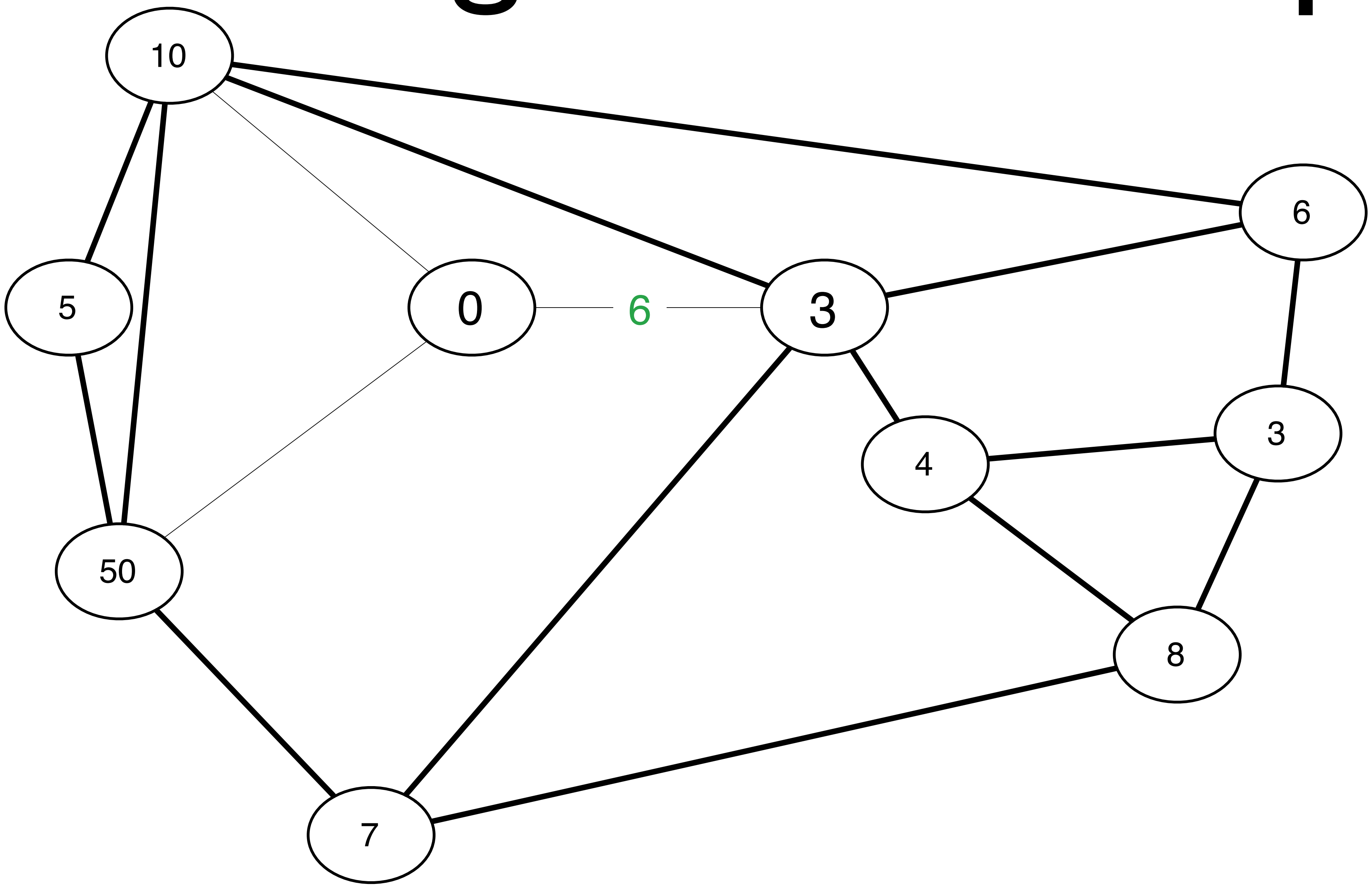
Reduce both endpoints' costs equally.

edge discount operation

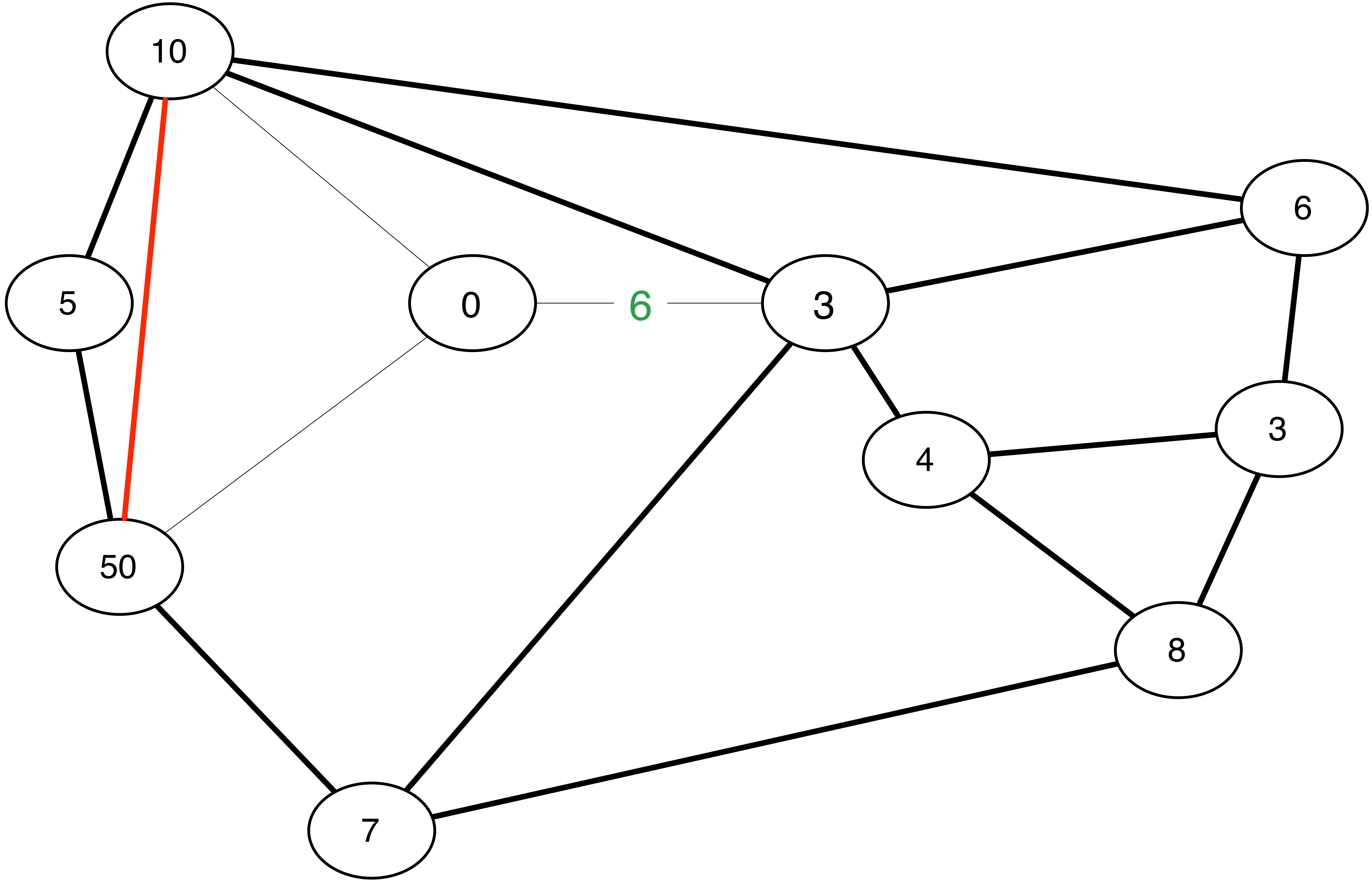


Reduce both endpoints' costs equally.

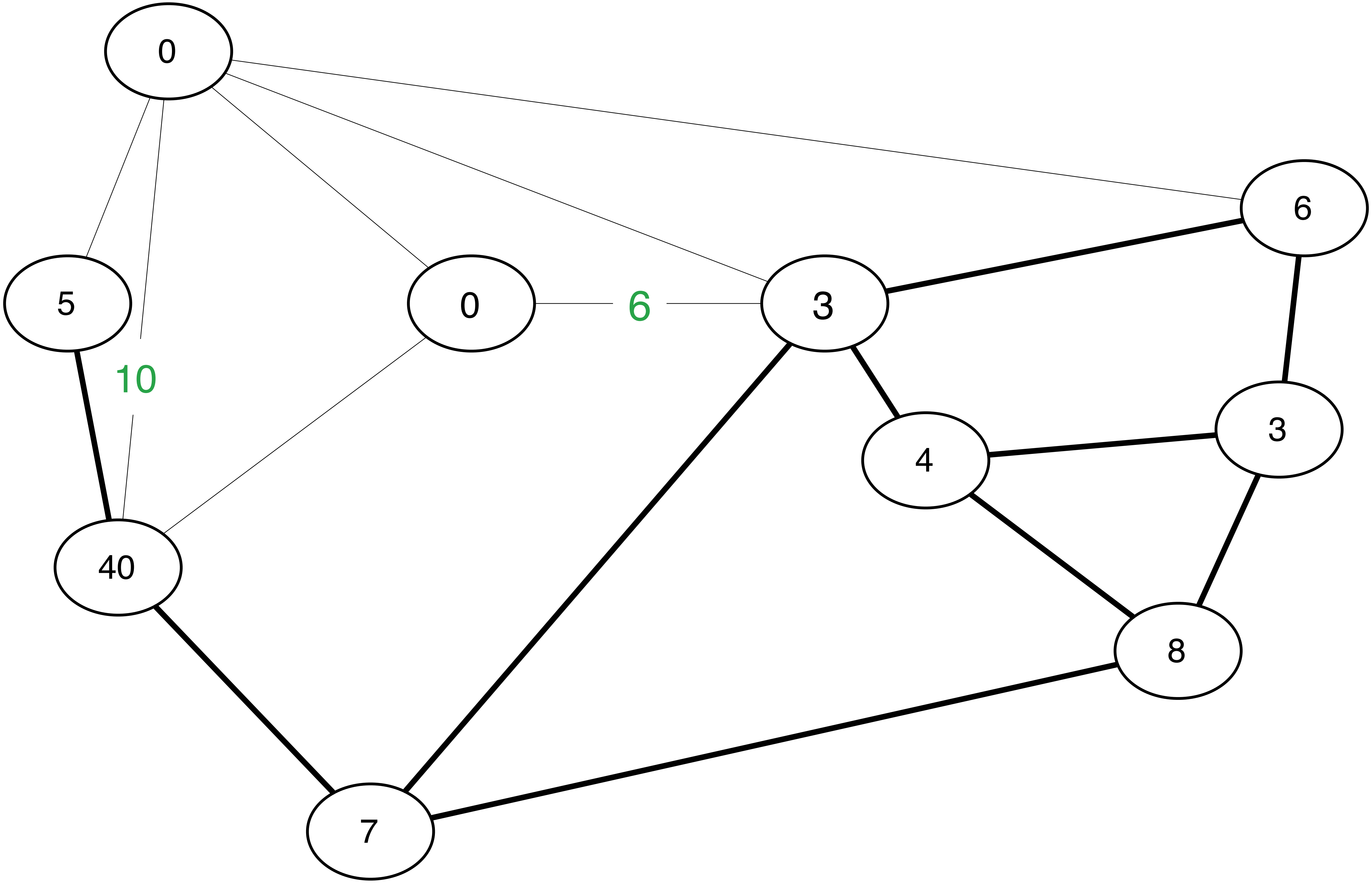
edge discount operation

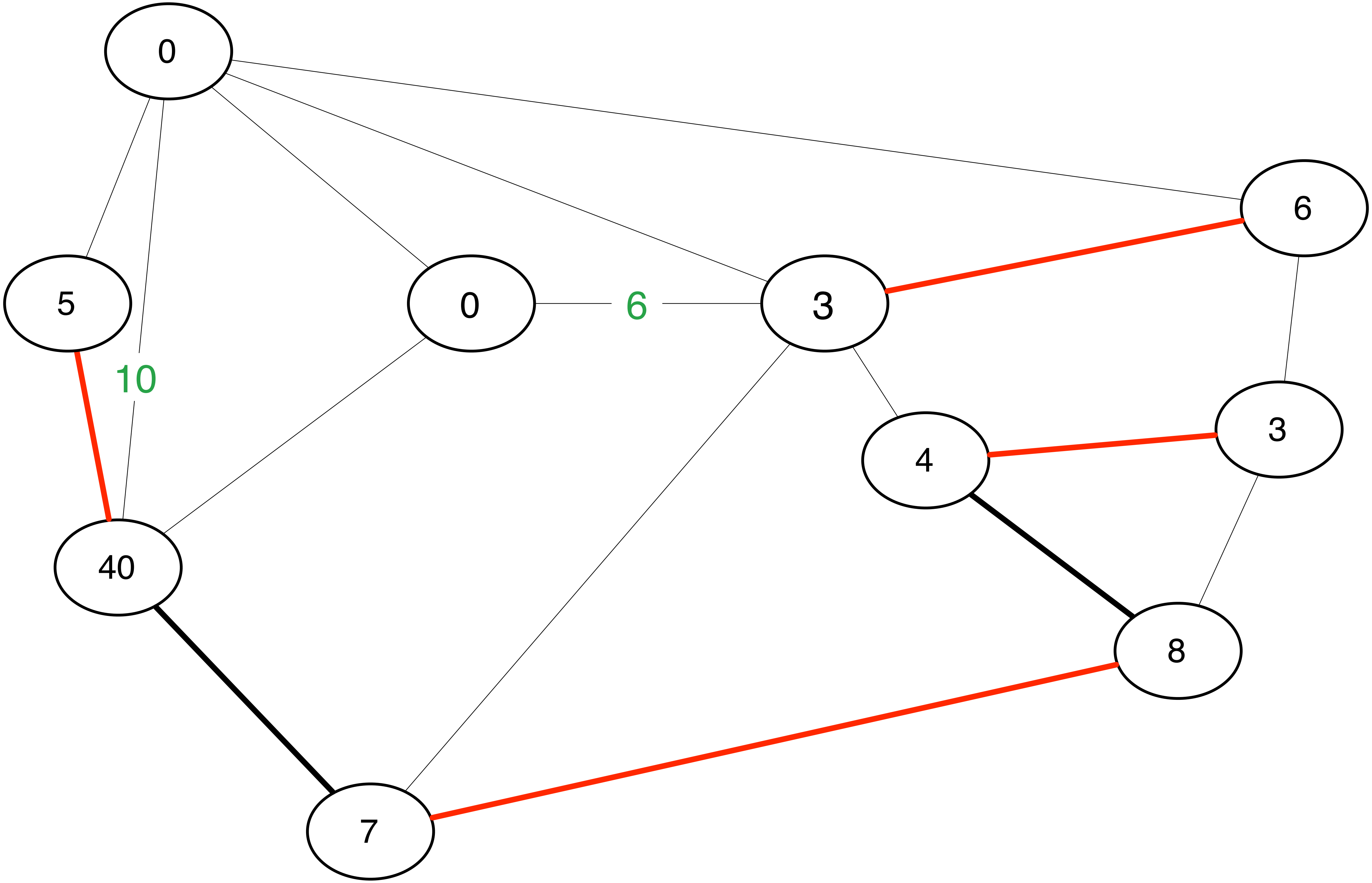


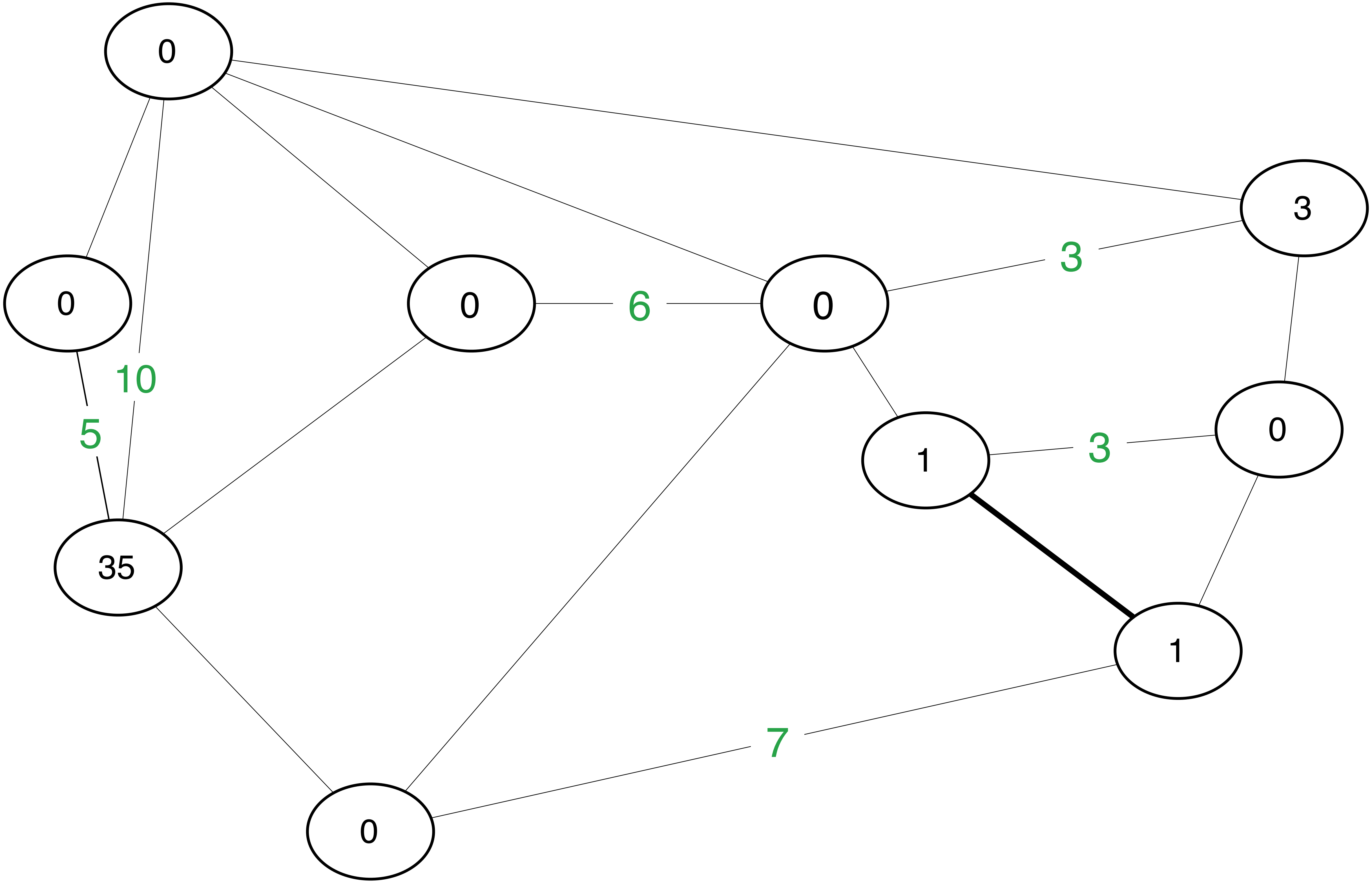
Reduce both endpoints' costs equally.

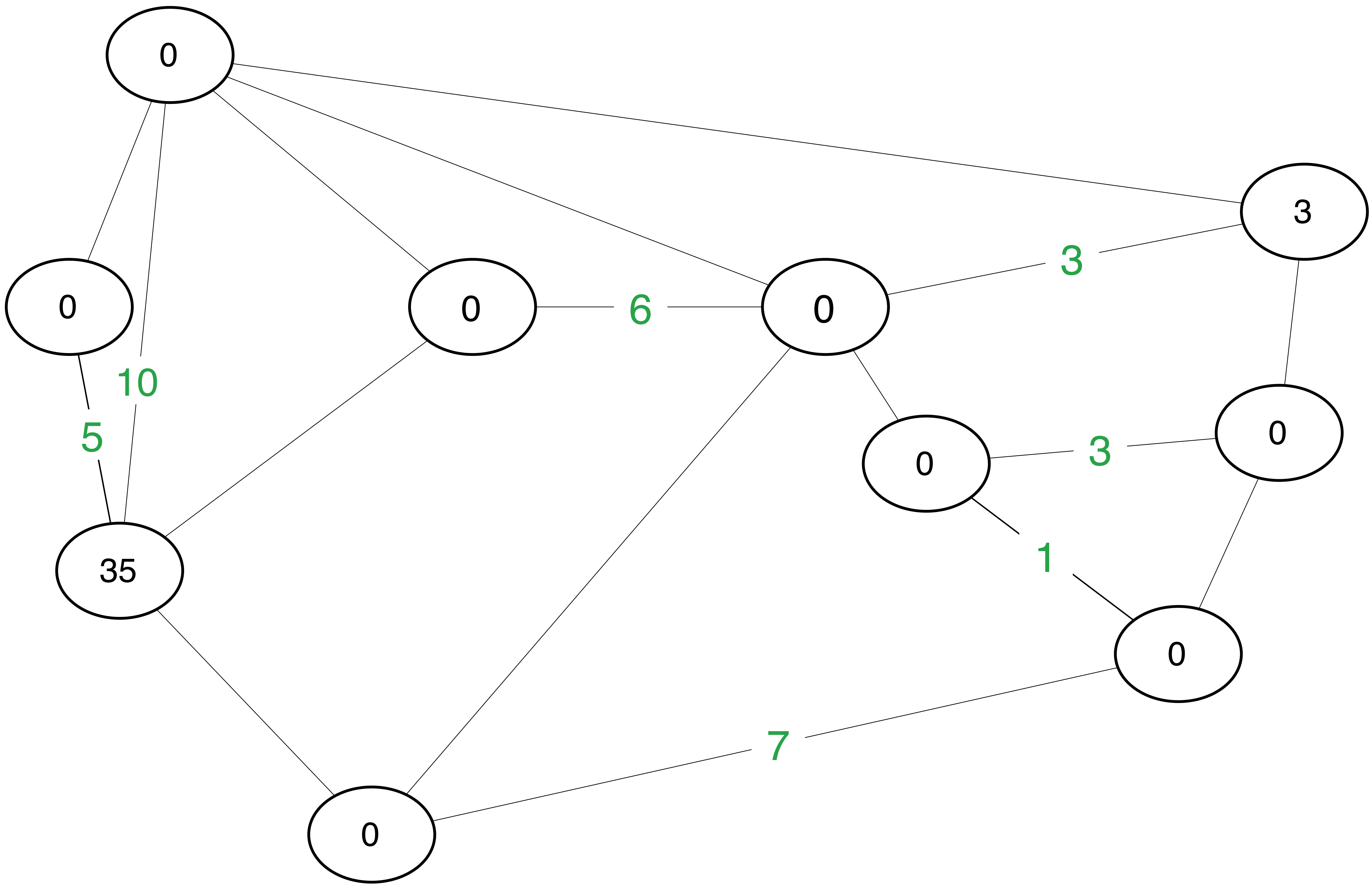


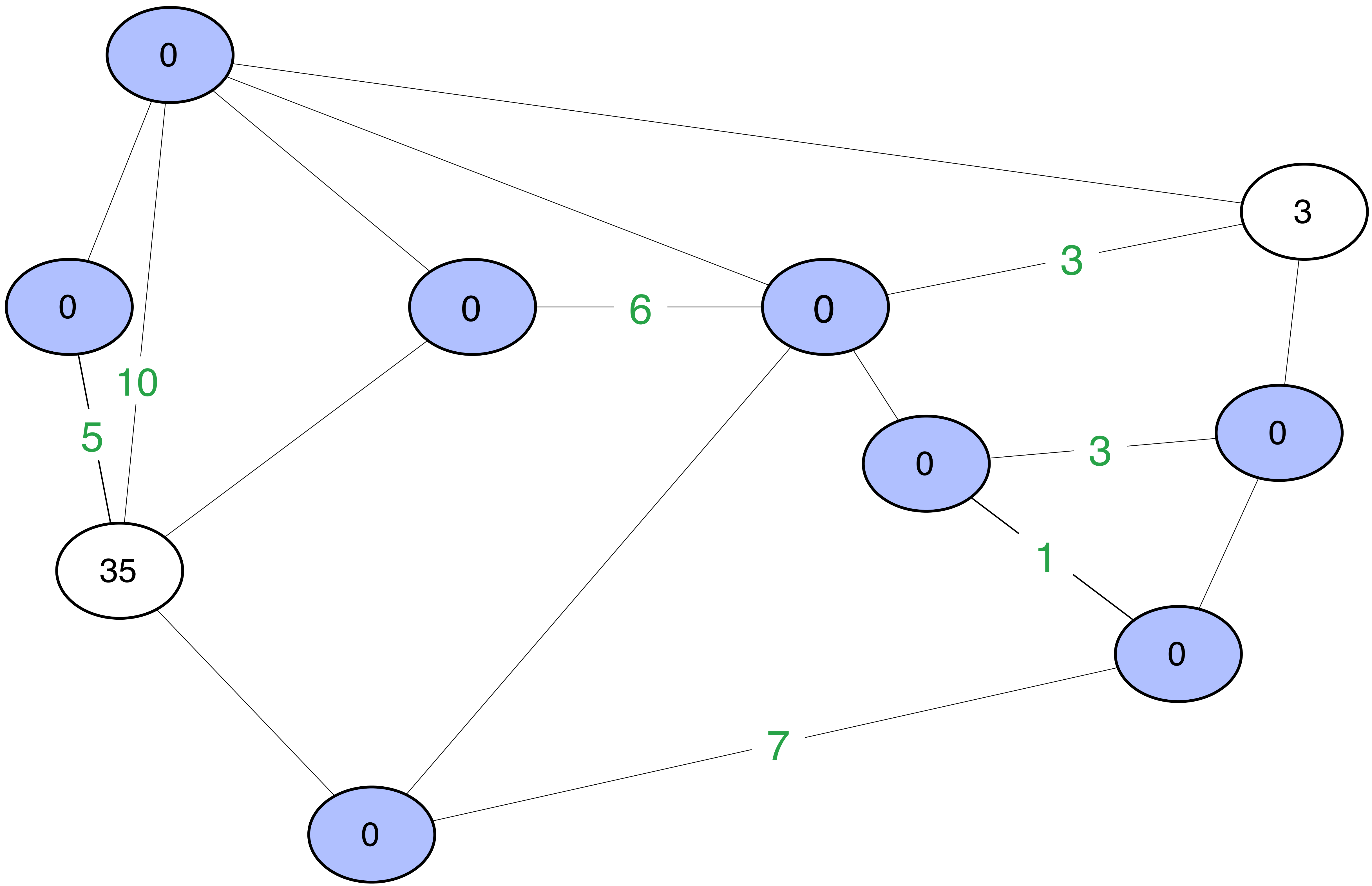
10
5
0
6
3
6
4
3
50
8
7

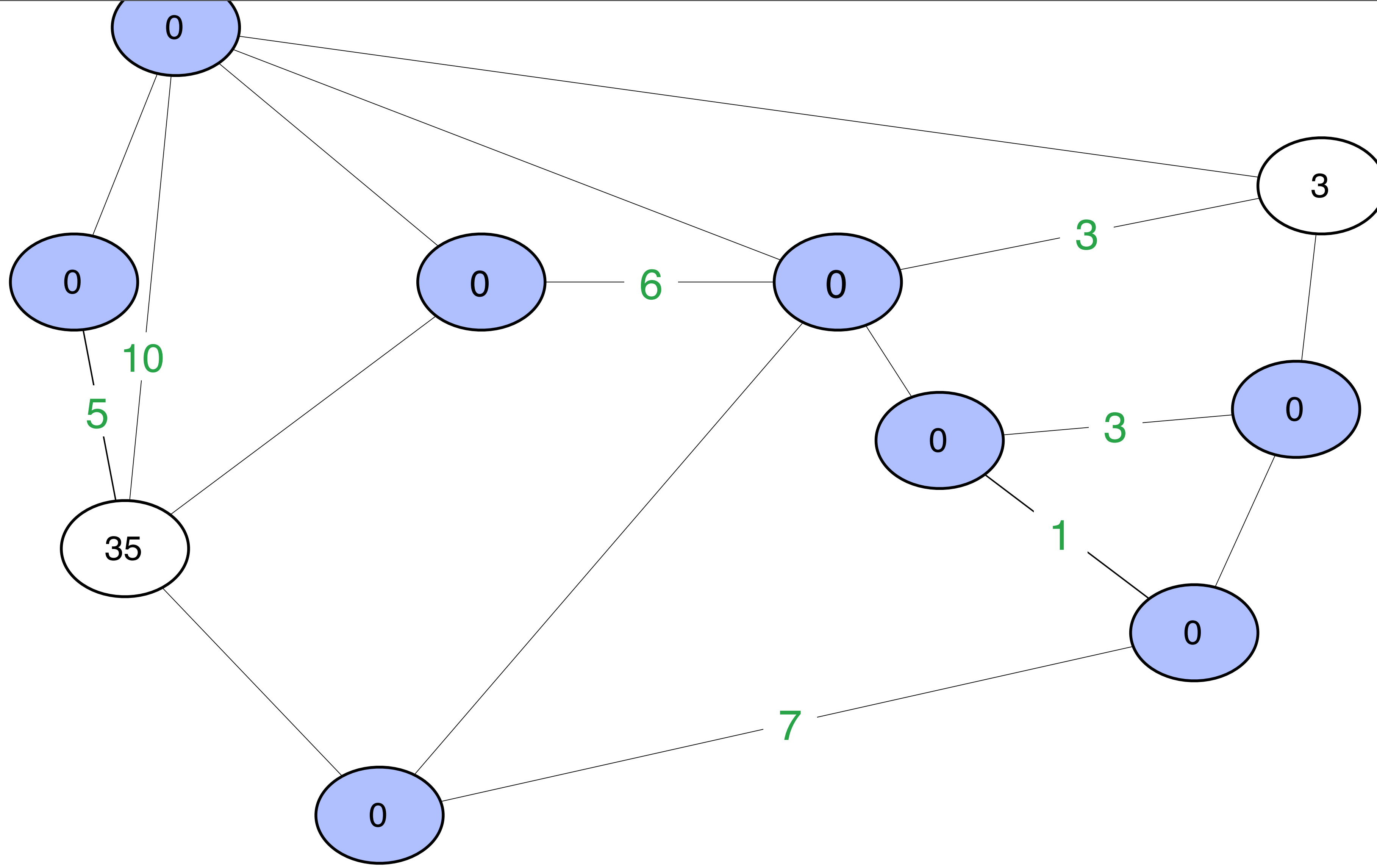

0
5
10
0
6
3
6
3
4
40
8
7


0
5
0
6
3
6
10
3
4
40
8
7


0
0
0
0
3
3
6
10
5
1
3
0
35
1
7
0


0
0
0
0
3
0
0
0
35
0
0
10
5
6
3
3
1
7


0
0
0
0
3
6
3
10
5
35
0
3
0
1
0
7
0

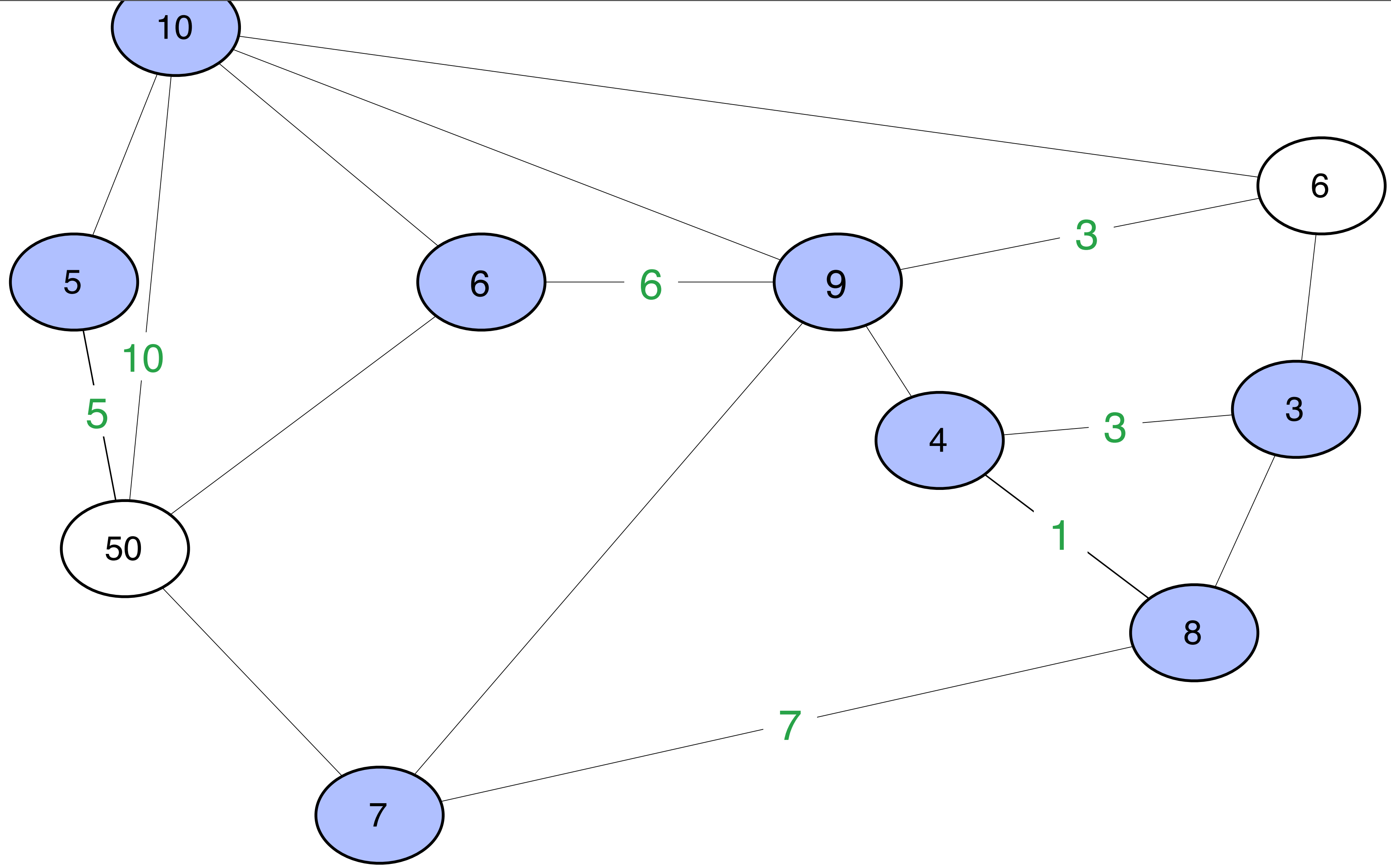


theorem: *cost(C) is at most* ***twice*** *the minimum possible cost.*

proof:

Show (i) *cost(C)* is at most twice the sum of the discounts.

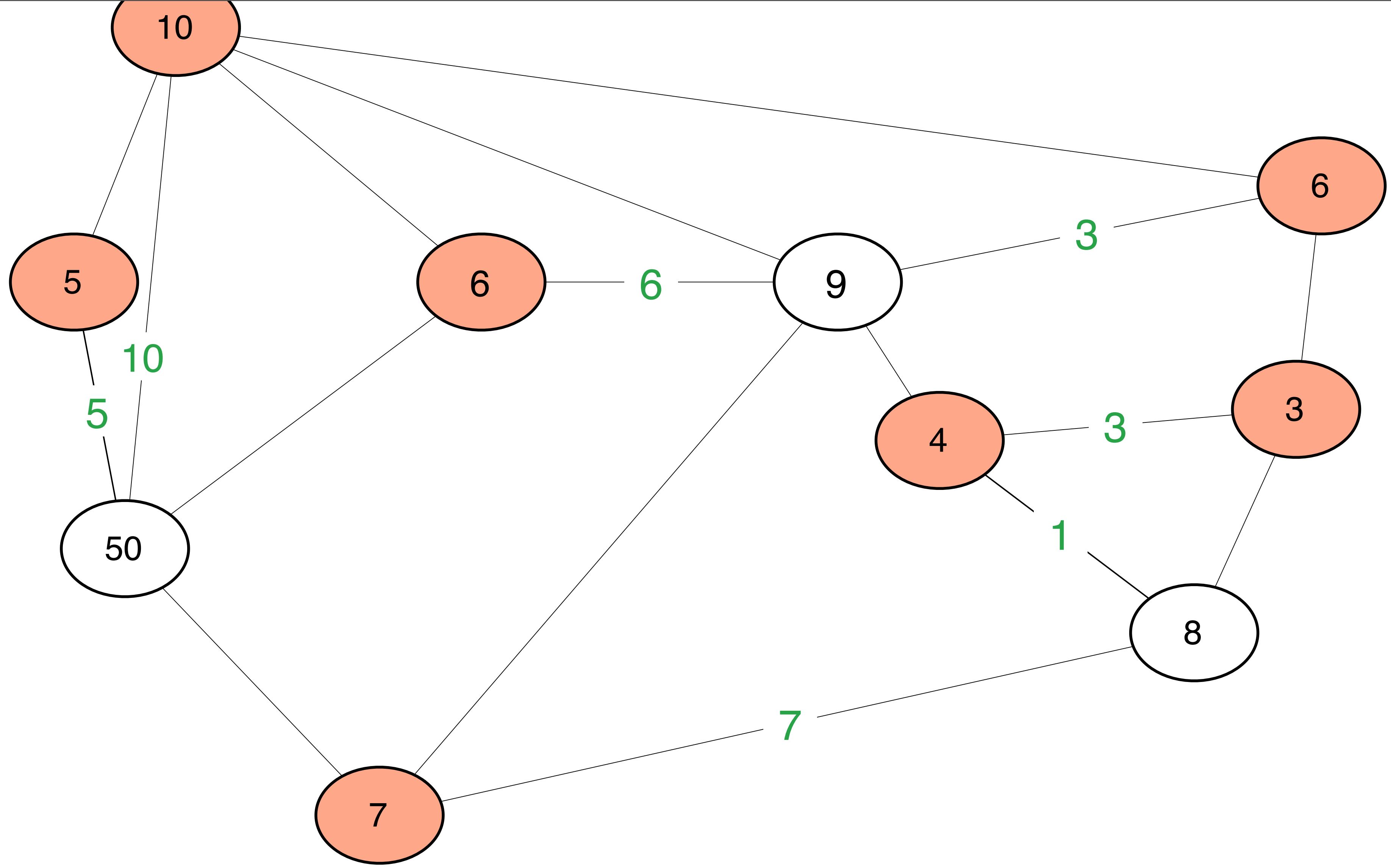
(ii) The sum of the discounts is at most the optimal cost.



step (i): *cost(C) is at most twice the sum of the discounts.*

proof:

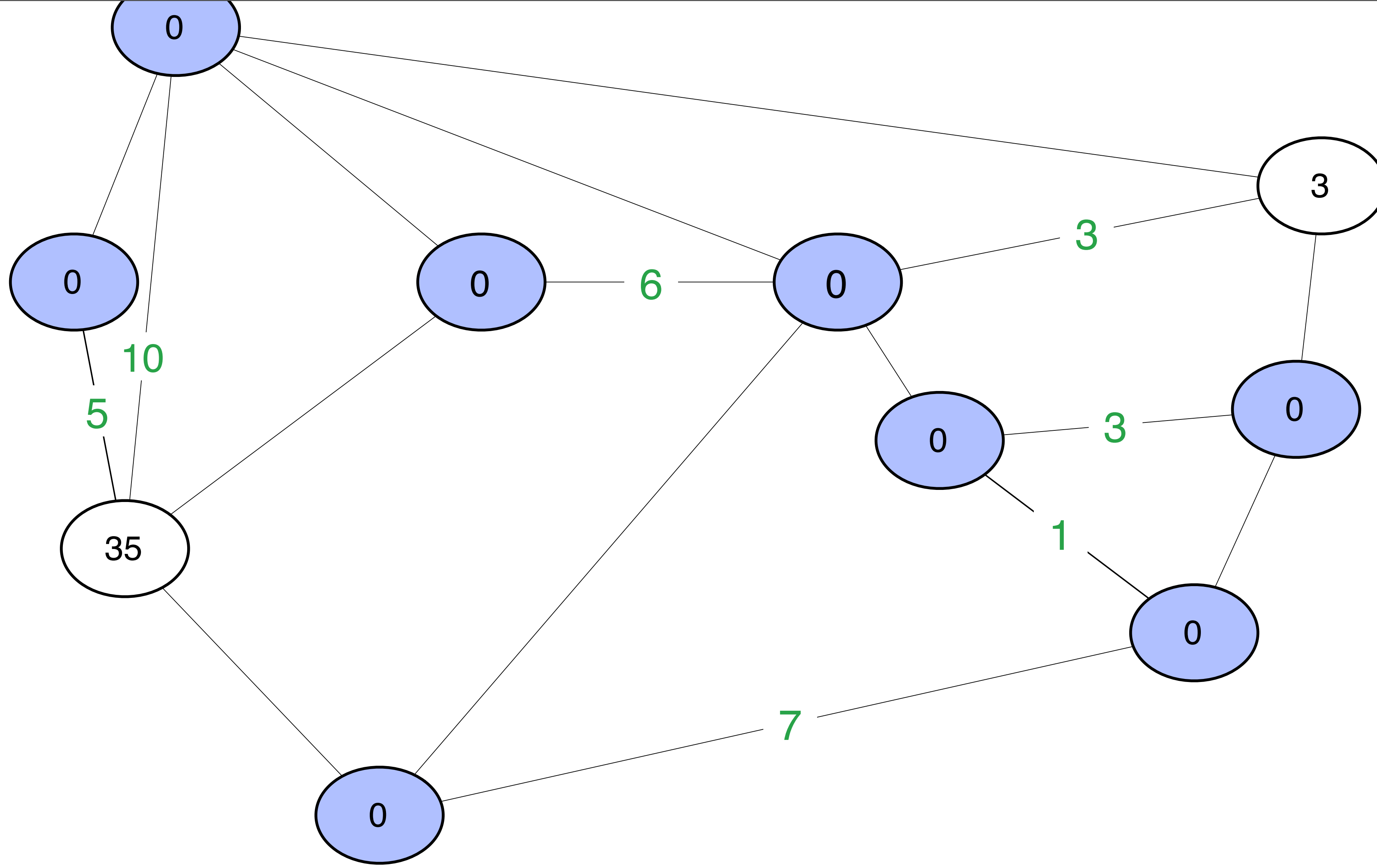
$$\sum_{v \in C} \mathrm{cost}(v) \;=\; \sum_{v \in C} \sum_{e \sim v} \mathrm{discount}(e) \;\le\; \sum_{e \in E} 2\,\mathrm{discount}(e)$$



step (ii): *Optimal cost is at least the sum of the discounts.*

proof:

$$\sum_{v \in \mathrm{OPT}} \mathrm{cost}(v) \;\ge\; \sum_{v \in \mathrm{OPT}} \sum_{e \sim v} \mathrm{discount}(e) \;\ge\; \sum_{e \in E} \mathrm{discount}(e)$$



theorem: *cost(C) is at most* ***twice*** *the minimum possible cost.*

proof:

(i) *cost(C)* is at most twice the sum of the discounts.

(ii) The sum of the discounts is at most the optimal cost.

next: fast *distributed* implementation of BY&E algorithm

each node knows only its neighbors

distributed computation

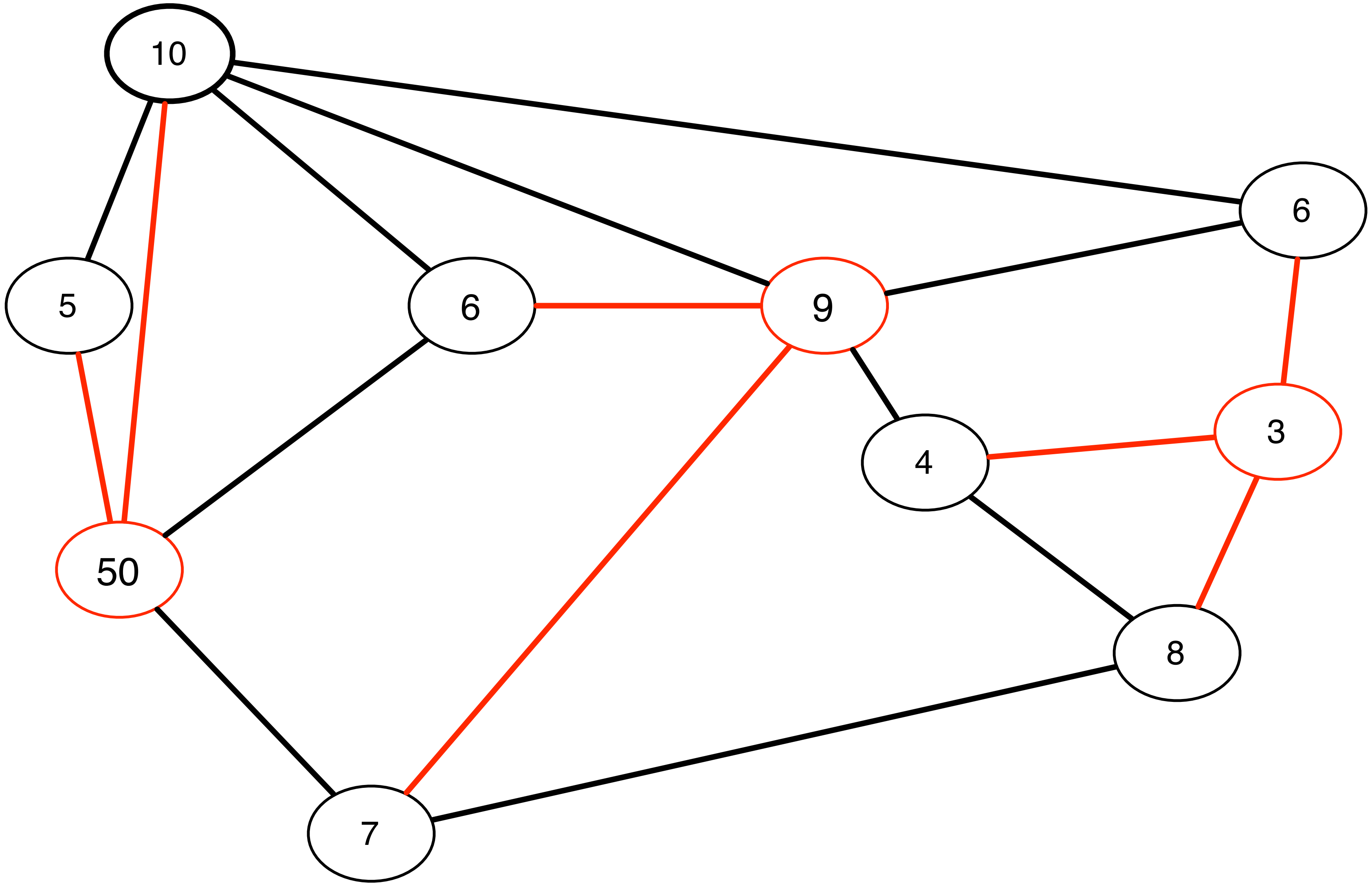
- Proceed in rounds.
- In each round:

  Each node exchanges O(1) messages

  with immediate neighbors,

  then does some computation.

goal: Finish in a small (logarithmic) number of rounds.



Each round:

1. form independent rooted "stars"
2. coordinate discounts within stars

Done when zero-cost vertices cover all edges.

goal: *Done after O(log n) rounds (n = #nodes).*

how to form stars

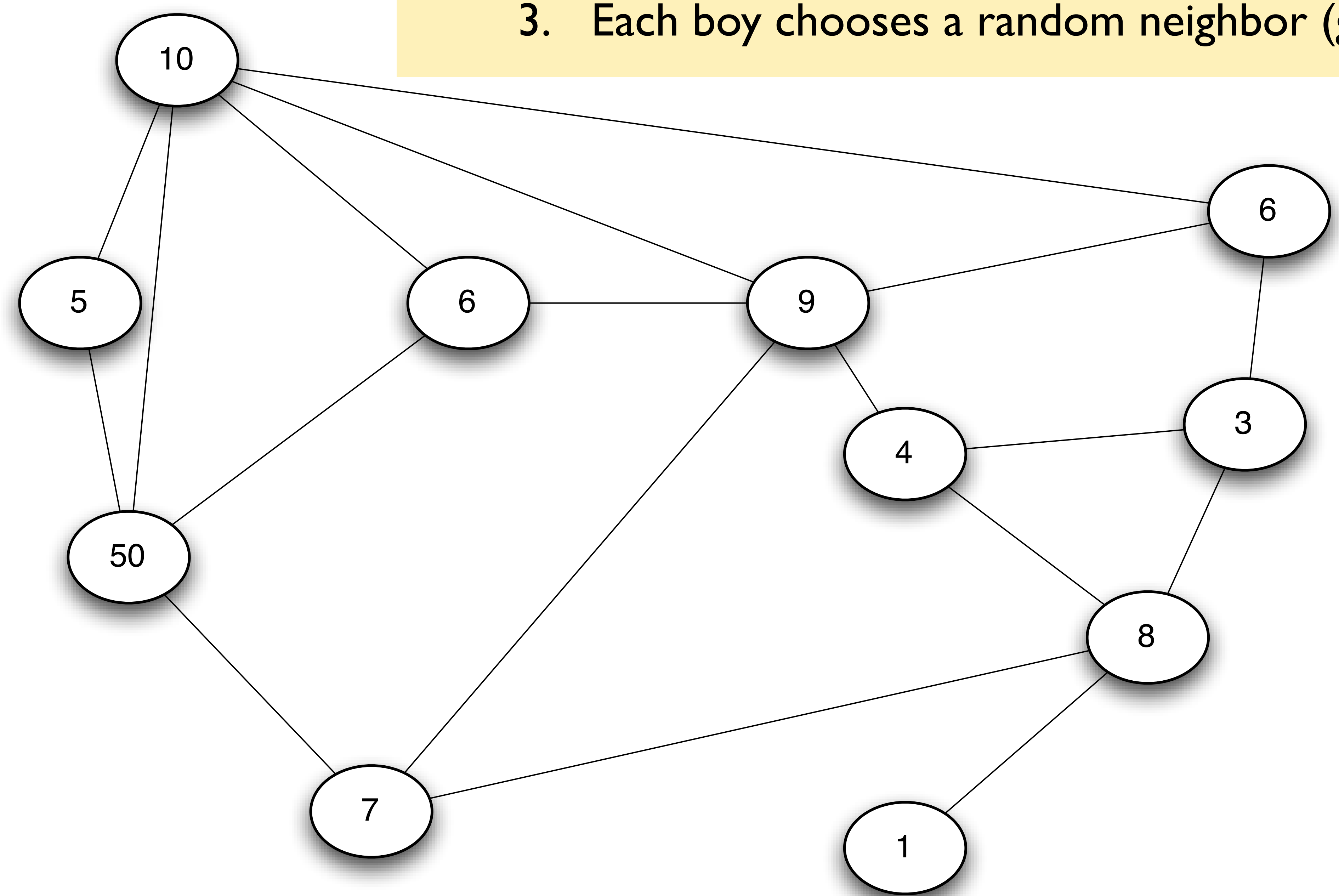
1. Each node randomly chooses to be "boy" or "girl" (just for this round).

2. For the round, use only edges from boys to higher-cost (or equal-cost) girls. (Pretend other edges don't exist.)†

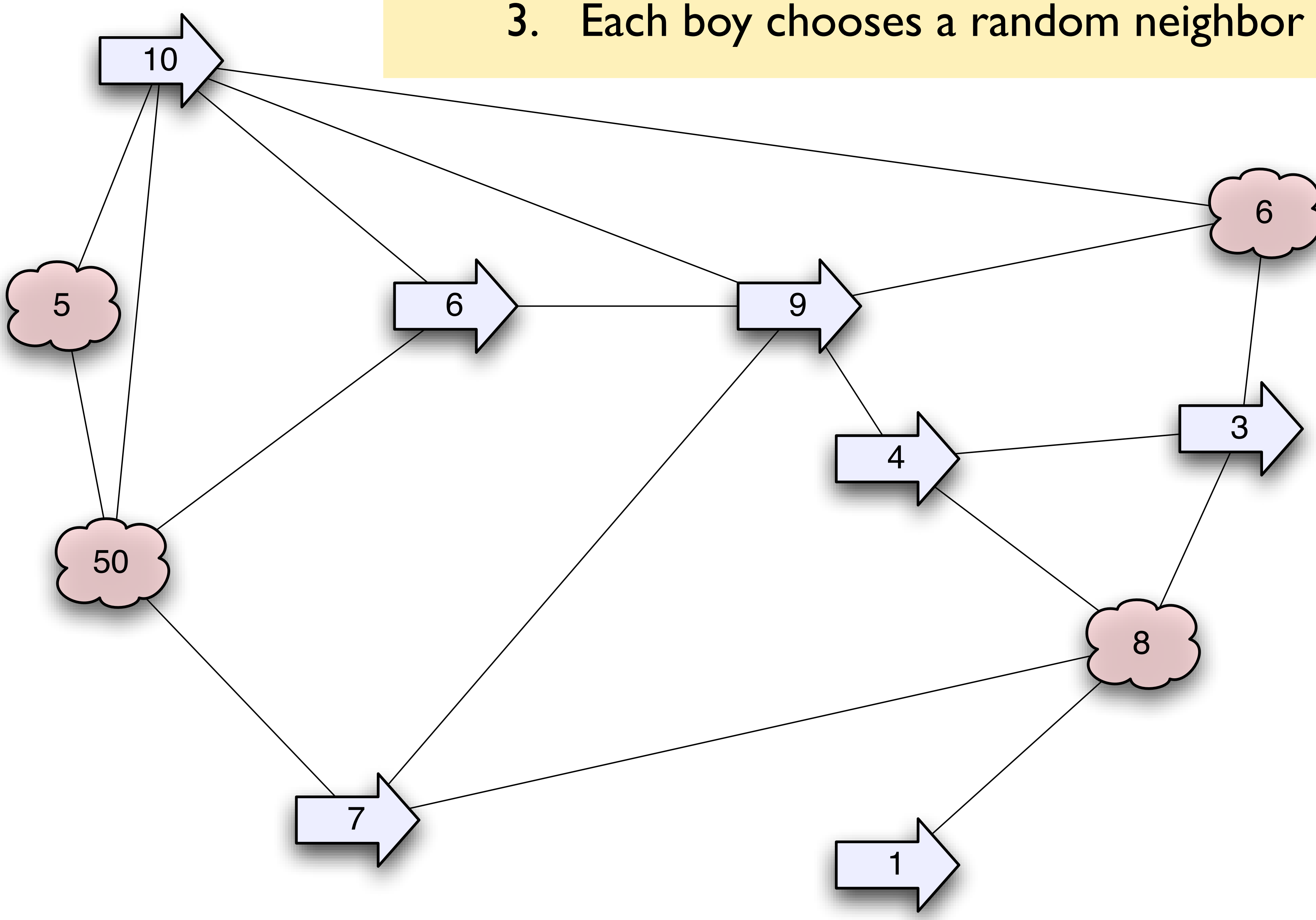
3. Each boy chooses a random neighbor (girl of $\geq$ cost).

† In each round, every edge has a one in four chance of being used.
(... will be used if low-cost endpoint is boy, high-cost endpoint is girl)

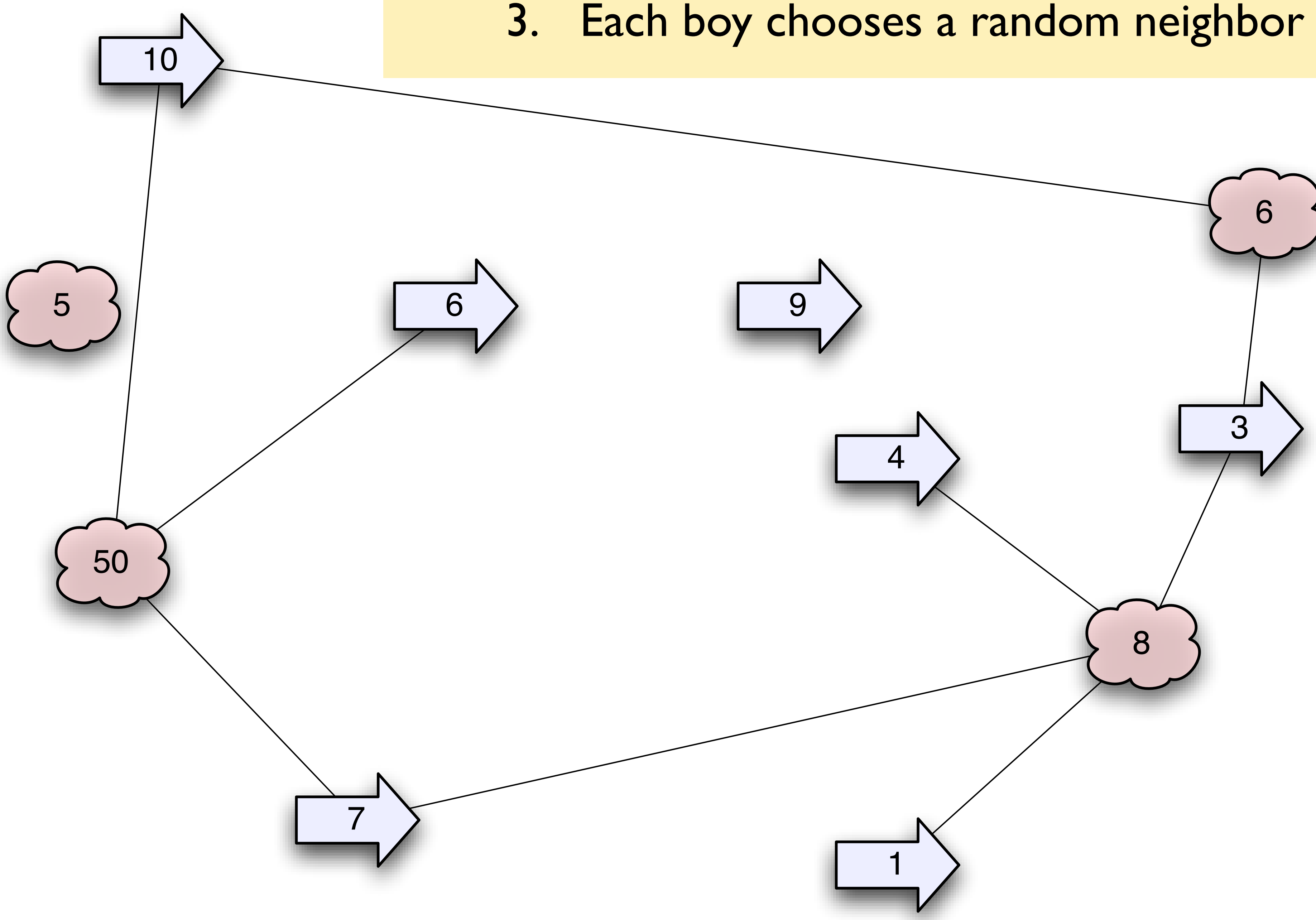
- How to form stars:
  1. Each node randomly chooses to be “boy” or “girl”.
  2. For the round, use only edges from boys to higher-cost (or equal-cost) girls. (Pretend other edges don’t exist.)
  3. Each boy chooses a random neighbor (girl of ≥ cost).



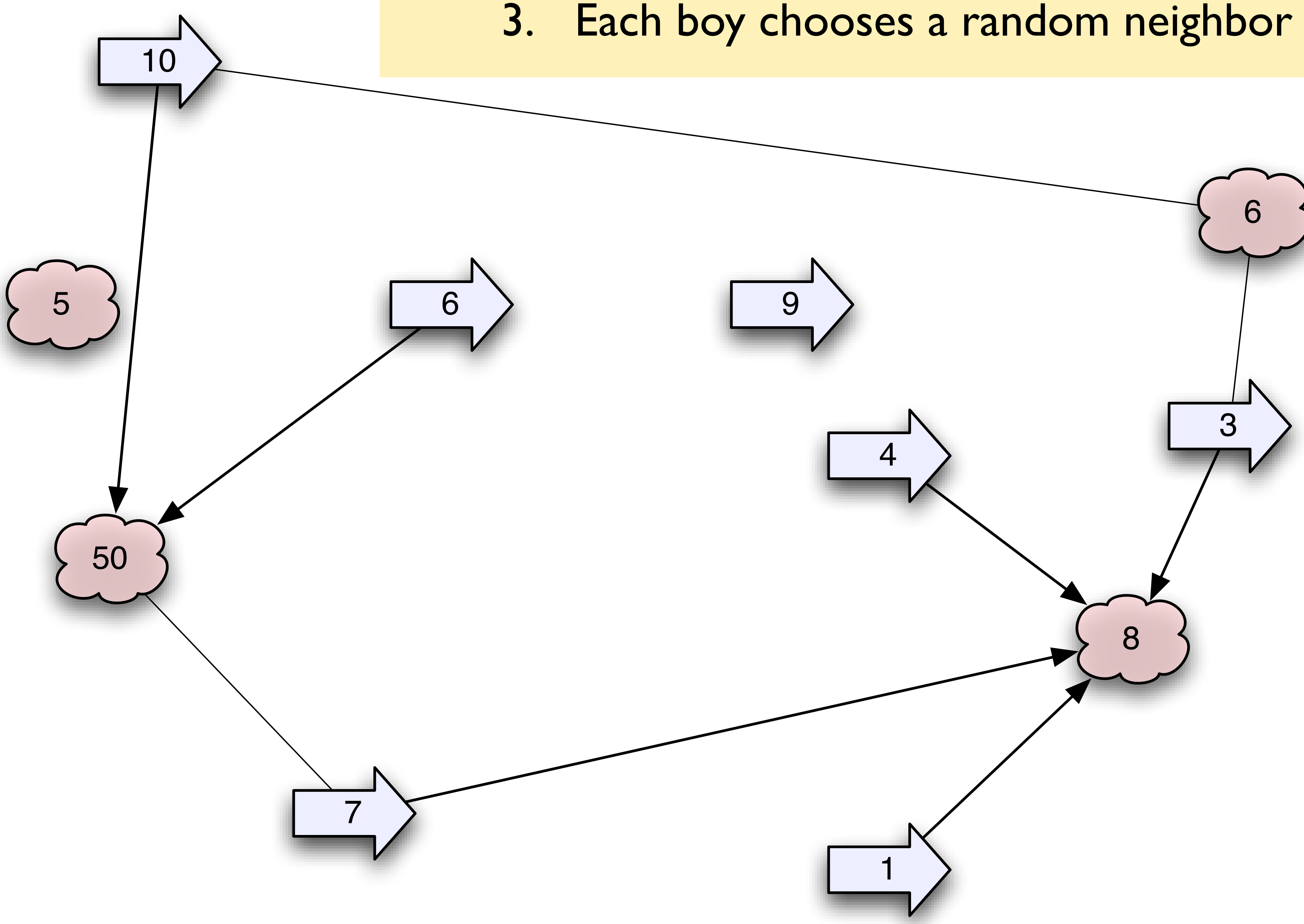
- How to form stars:
  1. Each node randomly chooses to be “boy” or “girl”.
  2. For the round, use only edges from boys to higher-cost (or equal-cost) girls. (Pretend other edges don’t exist.)
  3. Each boy chooses a random neighbor (girl of ≥ cost).



- How to form stars:
  1. Each node randomly chooses to be "boy" or "girl".
  2. For the round, use only edges from boys to higher-cost (or equal-cost) girls. (Pretend other edges don't exist.)
  3. Each boy chooses a random neighbor (girl of ≥ cost).



- How to form stars:
    1. Each node randomly chooses to be “boy” or “girl”.
    2. For the round, use only edges from boys to higher-cost (or equal-cost) girls. (Pretend other edges don’t exist.)
    3. Each boy chooses a random neighbor (girl of ≥ cost).



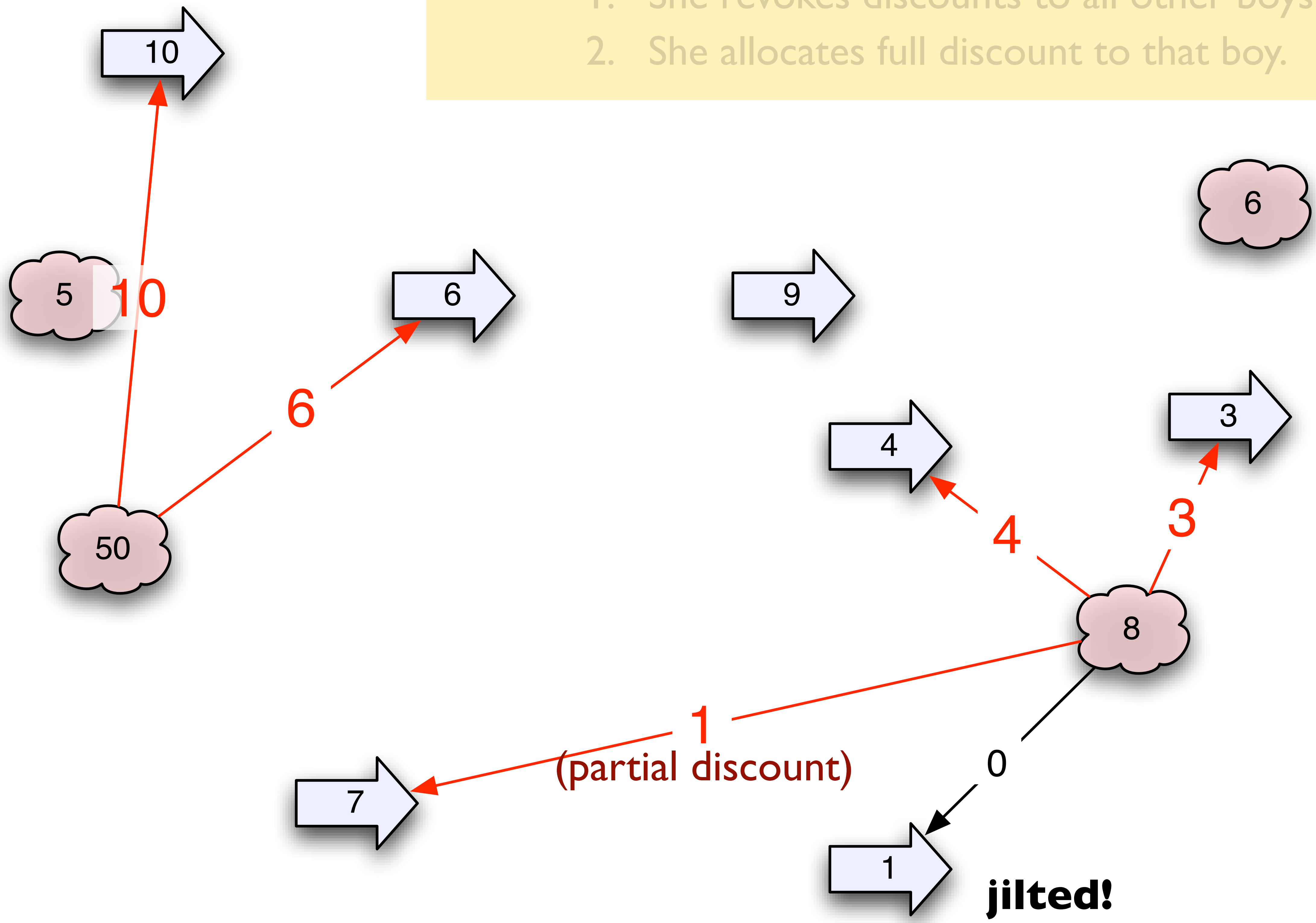
how girls allocate discounts

- Each girl allocates discounts greedily, in alphabetic order.
- If she *partially allocates* some boy's discount, then...

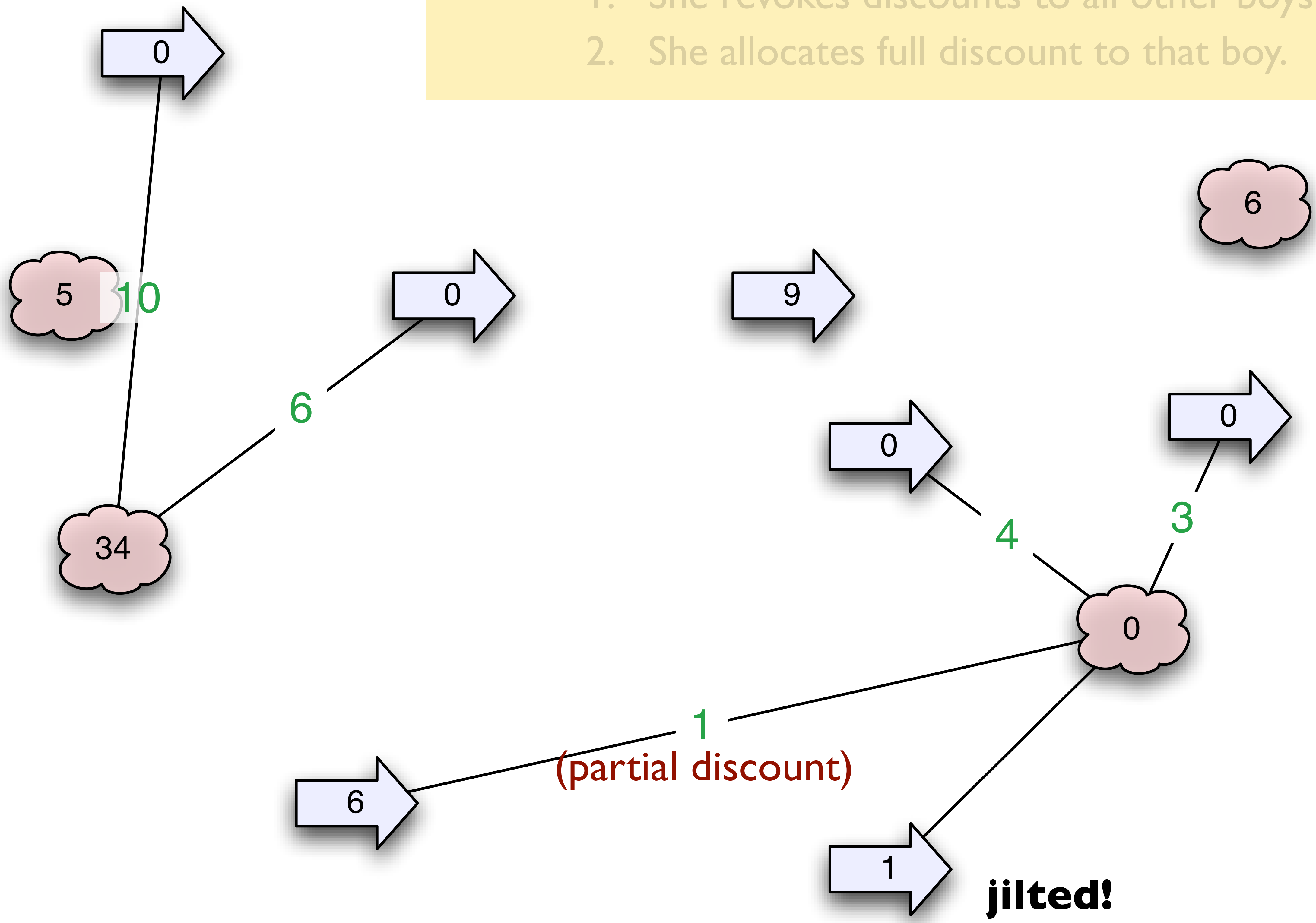
  with probability 1/2:

  1. She revokes discounts to all other boys.
  2. She allocates full discount to that boy.
- Some boys may be *jilted* (have *no chance* for discount).

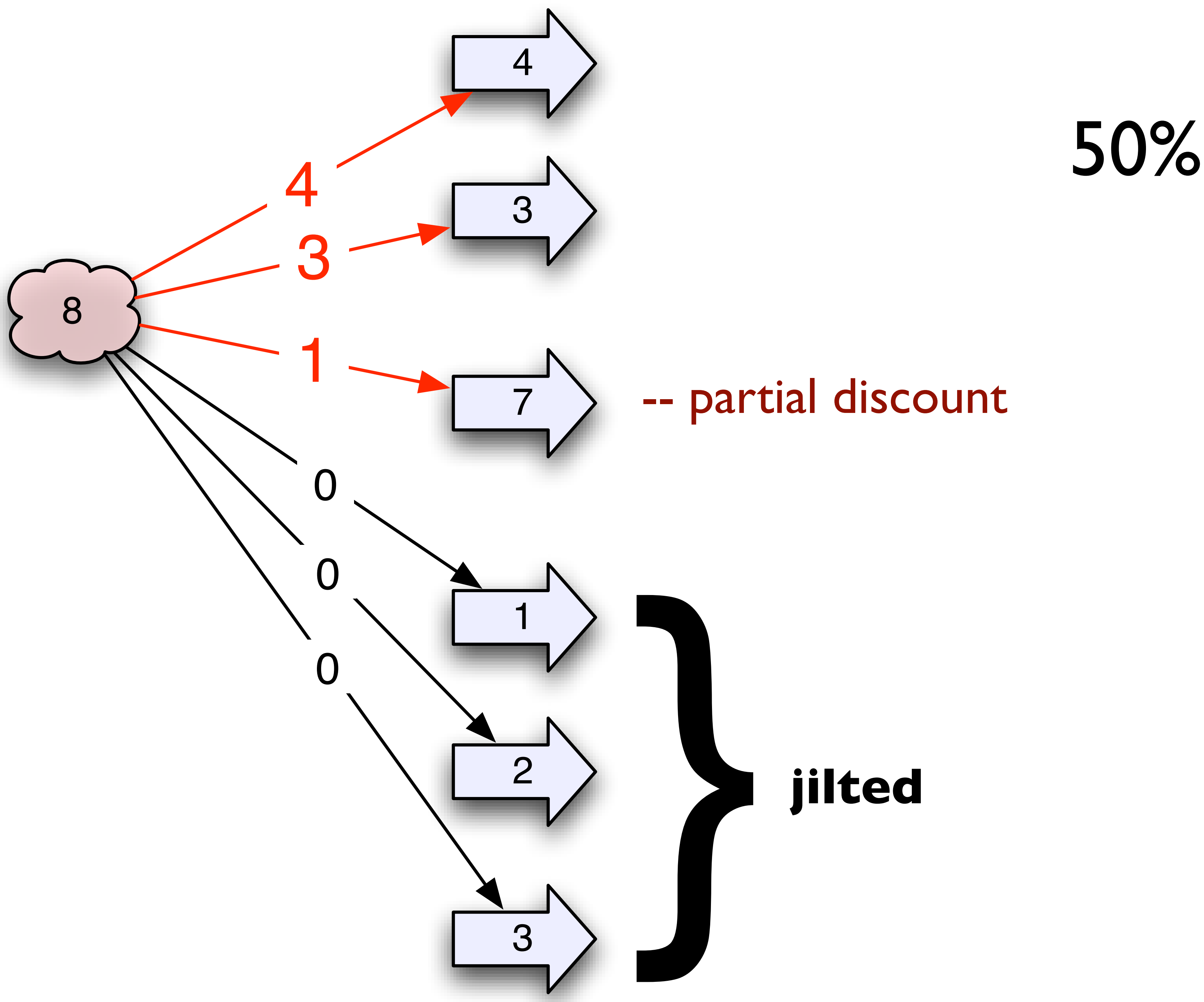
- Each girl allocates discounts greedily, in alphabetical order.
- If she *partially allocates* some boy's discount, then... with probability 1/2:
  1. She revokes discounts to all other boys.
  2. She allocates full discount to that boy.



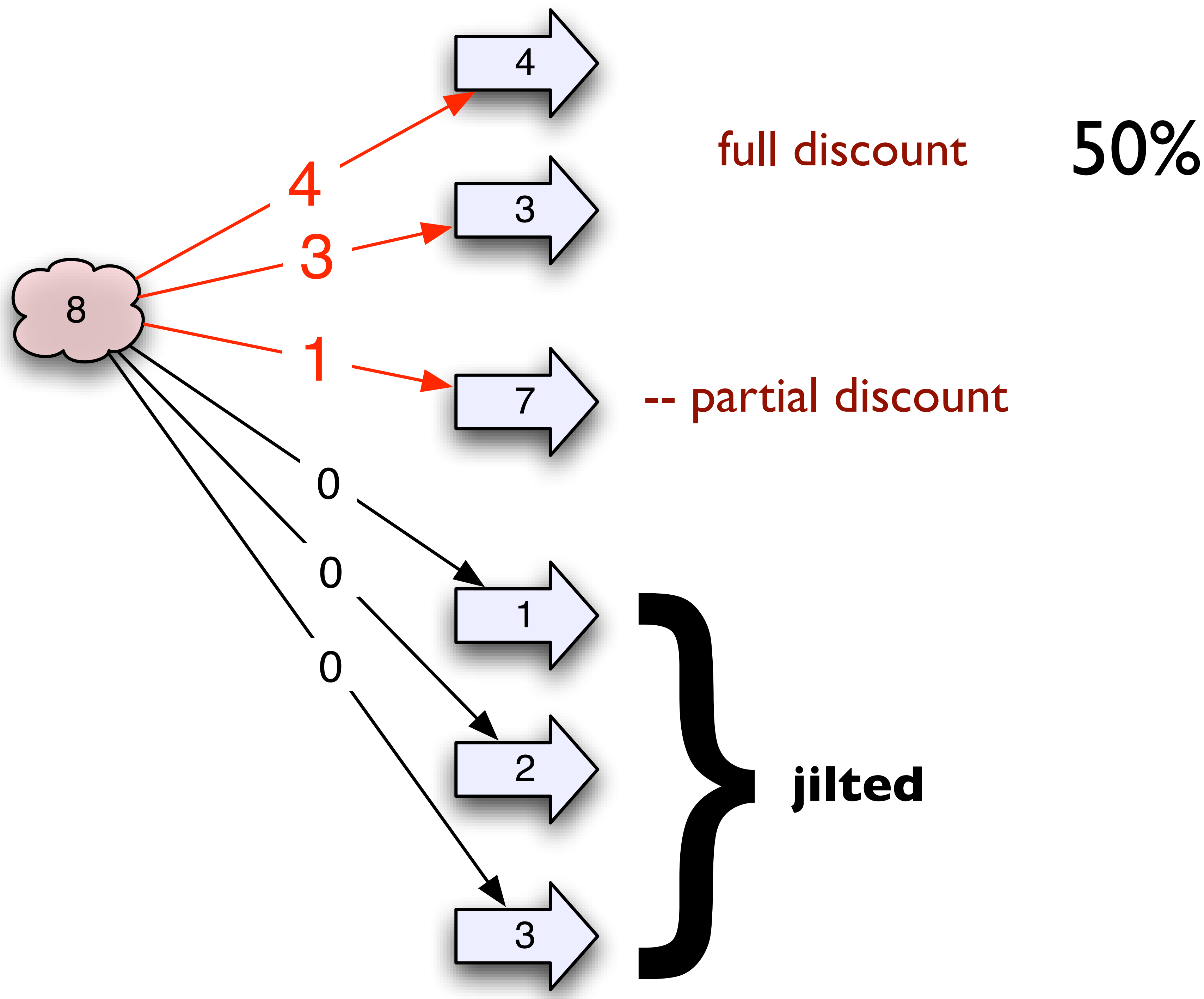
- Each girl allocates discounts greedily, in alphabetical order.
- If she *partially allocates* some boy's discount, then... with probability 1/2:
  1. She revokes discounts to all other boys.
  2. She allocates full discount to that boy.



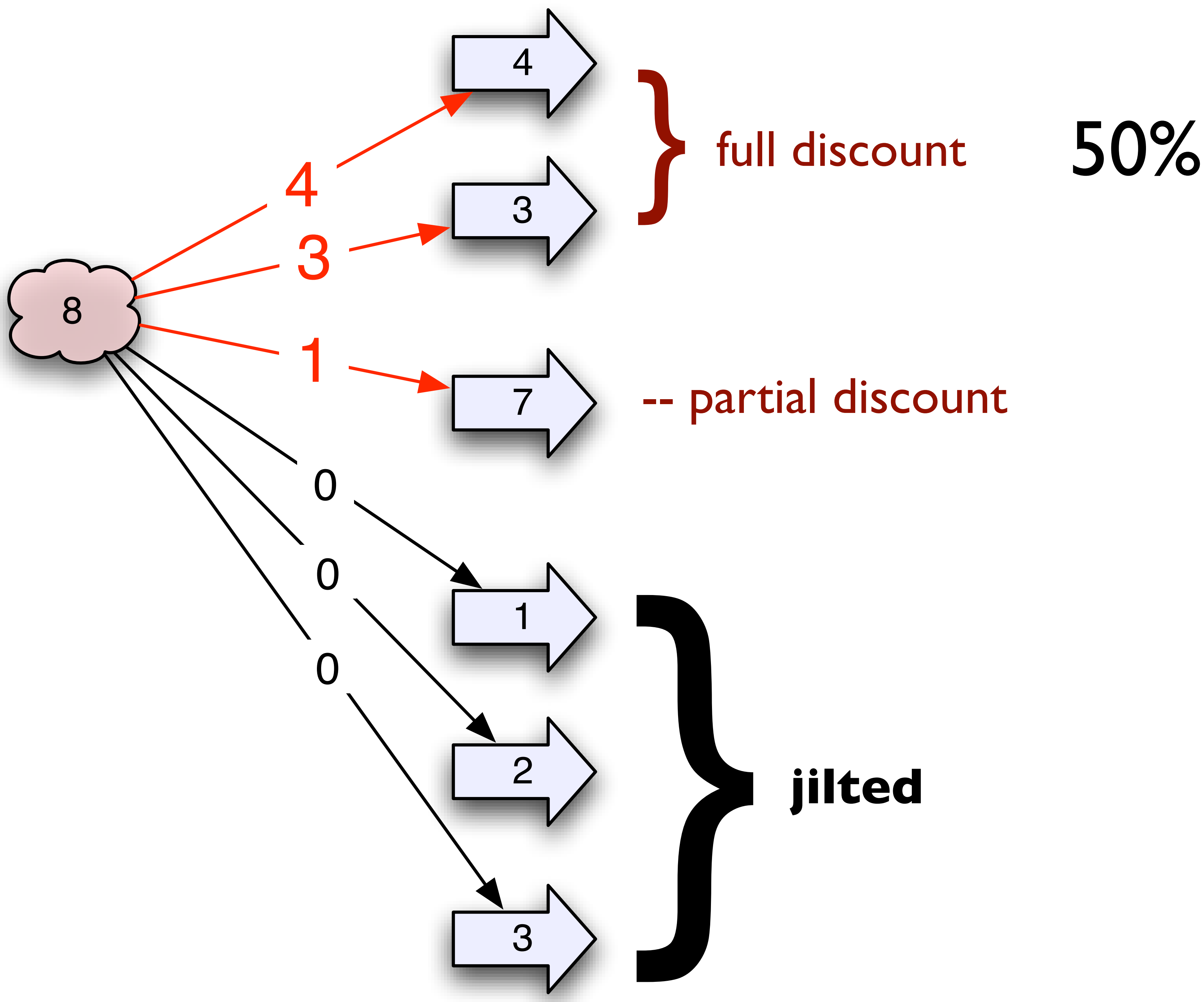
- Each girl allocates discounts greedily, in alphabetical order.
- If she *partially allocates* some boy's discount, then... with probability 1/2:
    1. She revokes discounts to all other boys.
    2. She allocates full discount to that boy.



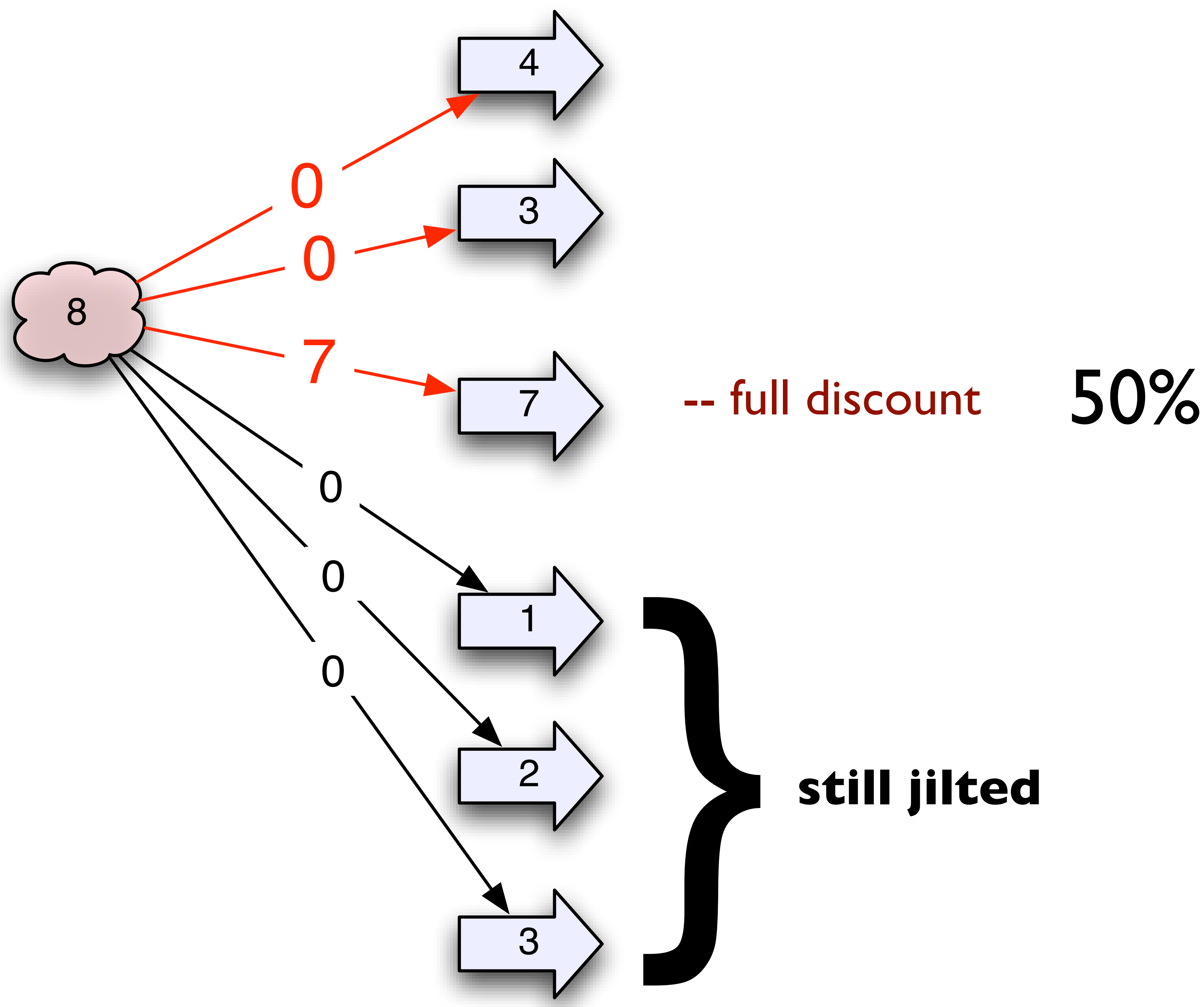
- Each girl allocates discounts greedily, in alphabetical order.
- If she *partially allocates* some boy's discount, then... with probability 1/2:
    1. She revokes discounts to all other boys.
    2. She allocates full discount to that boy.



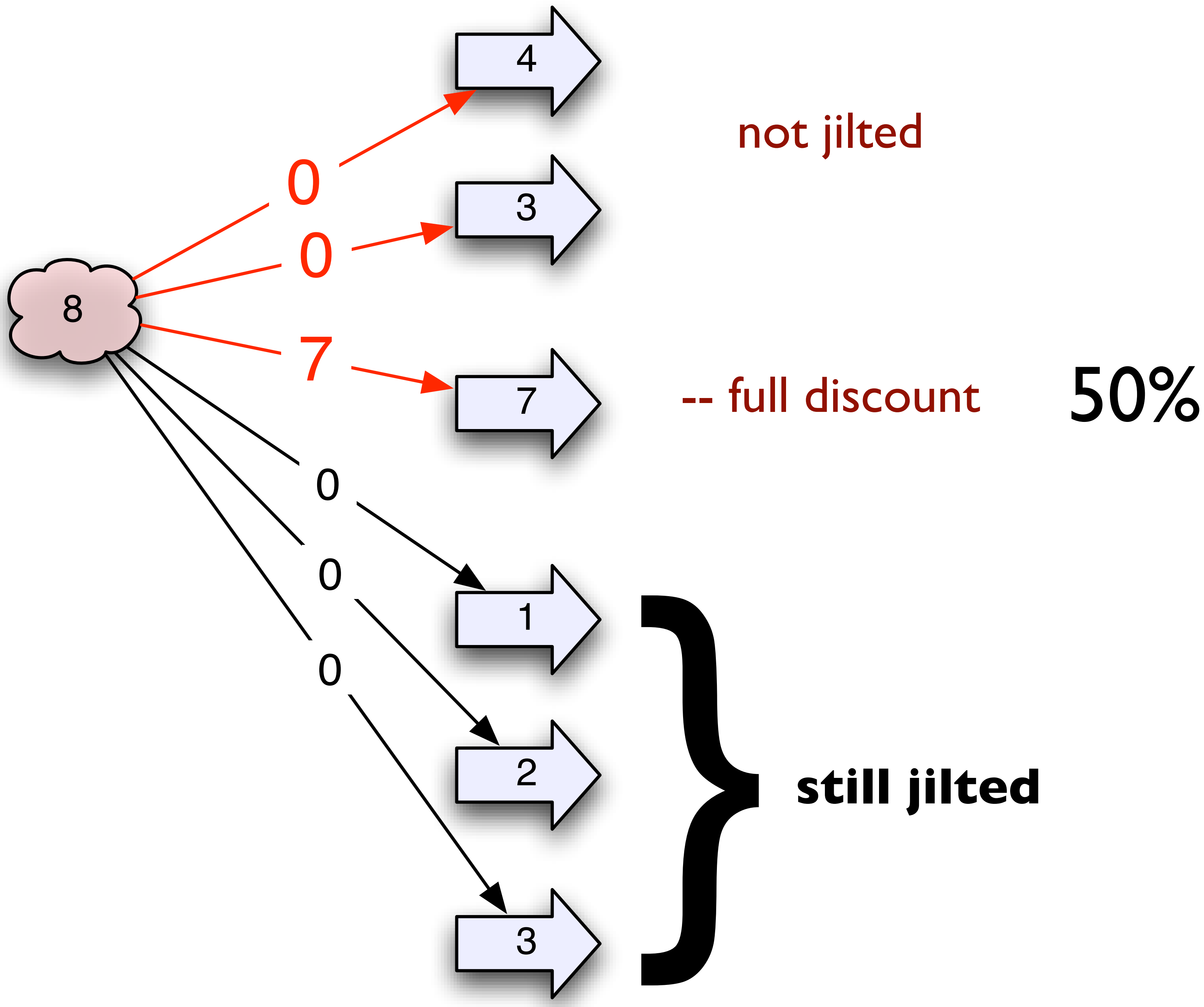
- Each girl allocates discounts greedily, in alphabetical order.
- If she *partially allocates* some boy's discount, then... with probability 1/2:
    1. She revokes discounts to all other boys.
    2. She allocates full discount to that boy.



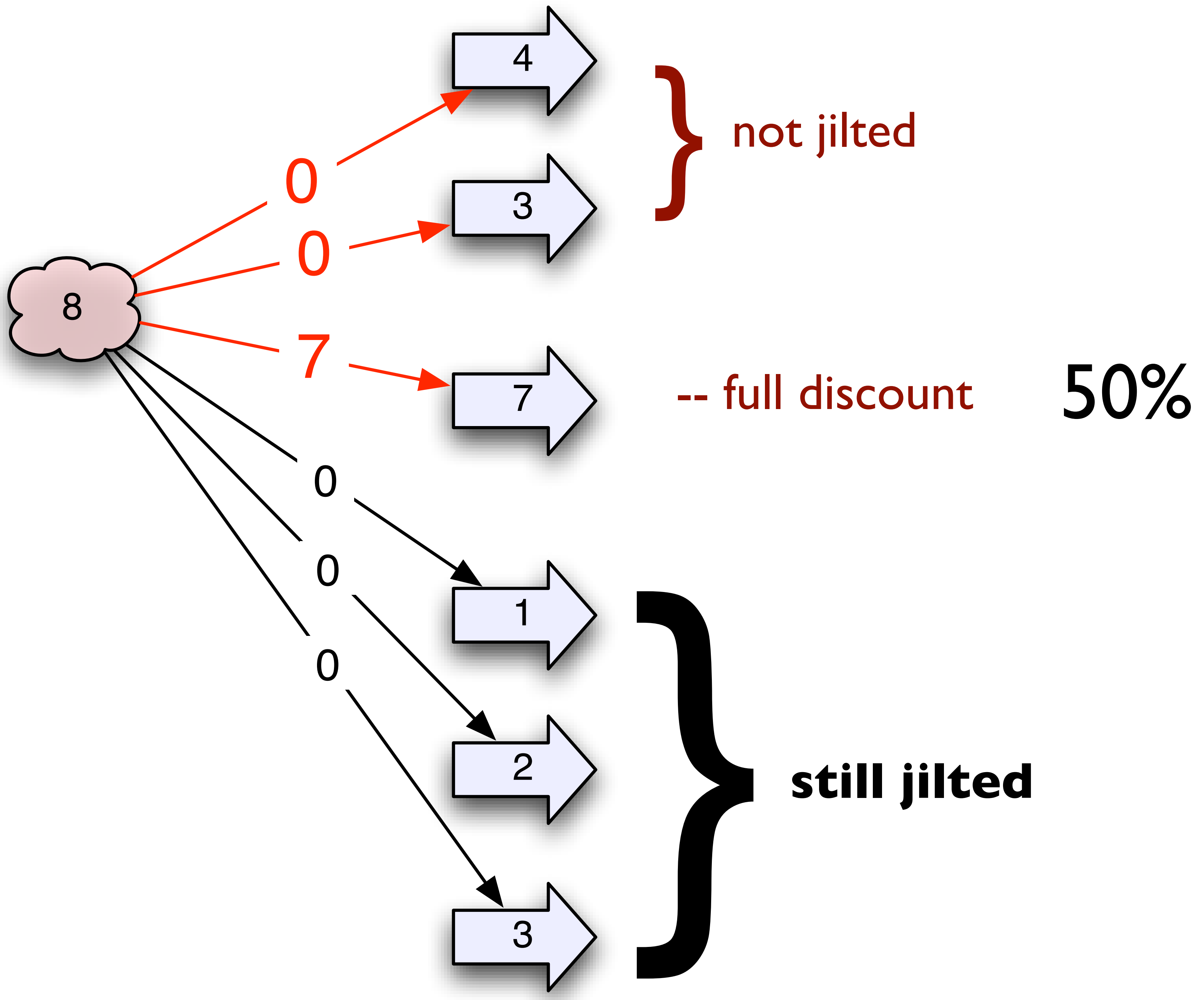
- Each girl allocates discounts greedily, in alphabetical order.
- If she *partially allocates* some boy's discount, then... with probability 1/2:
    1. She revokes discounts to all other boys.
    2. She allocates full discount to that boy.



- Each girl allocates discounts greedily, in alphabetical order.
- If she *partially allocates* some boy's discount, then... with probability 1/2:
  1. She revokes discounts to all other boys.
  2. She allocates full discount to that boy.

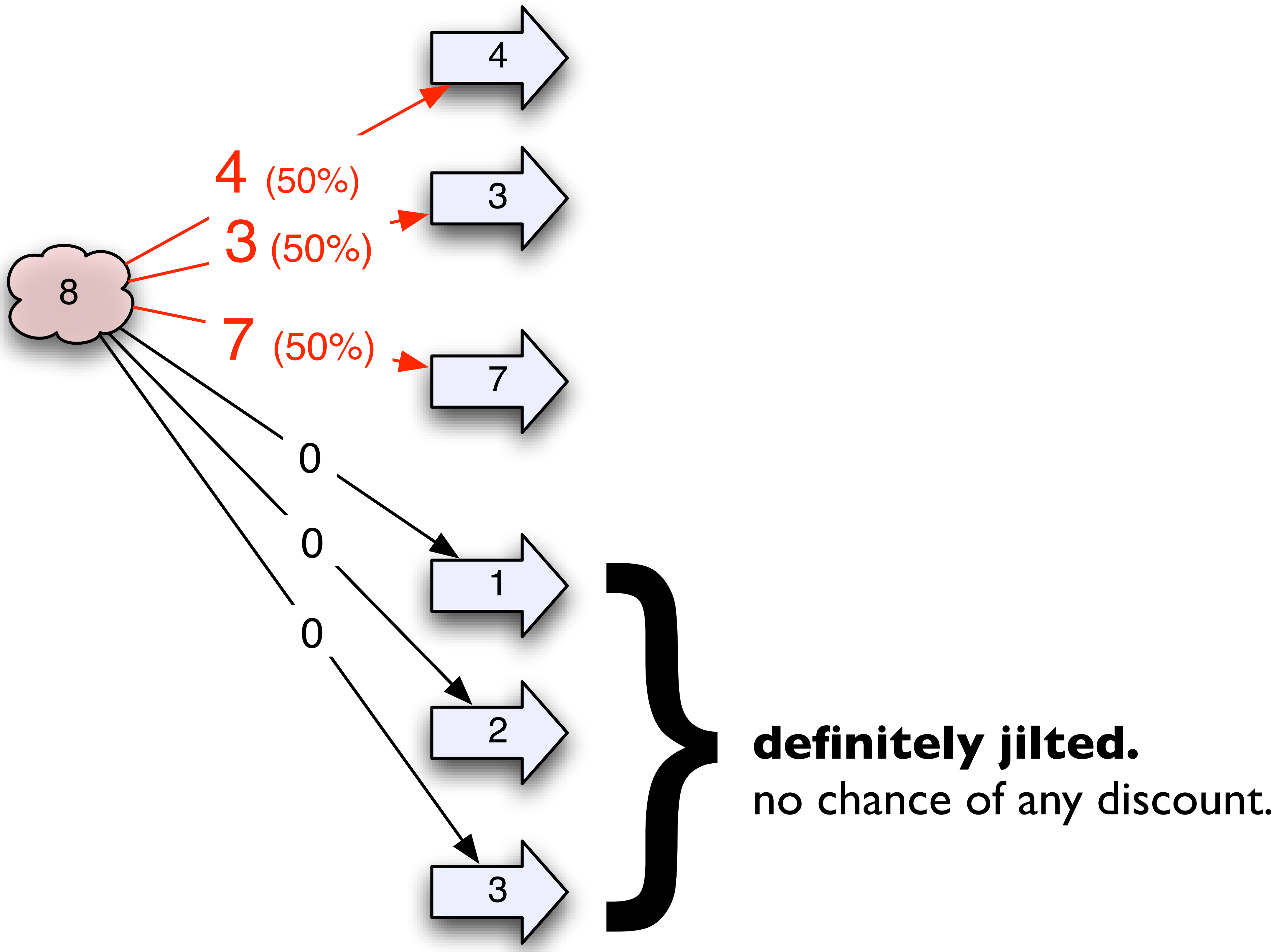


- Each girl allocates discounts greedily, in alphabetical order.
- If she *partially allocates* some boy's discount, then... with probability 1/2:
    1. She revokes discounts to all other boys.
    2. She allocates full discount to that boy.



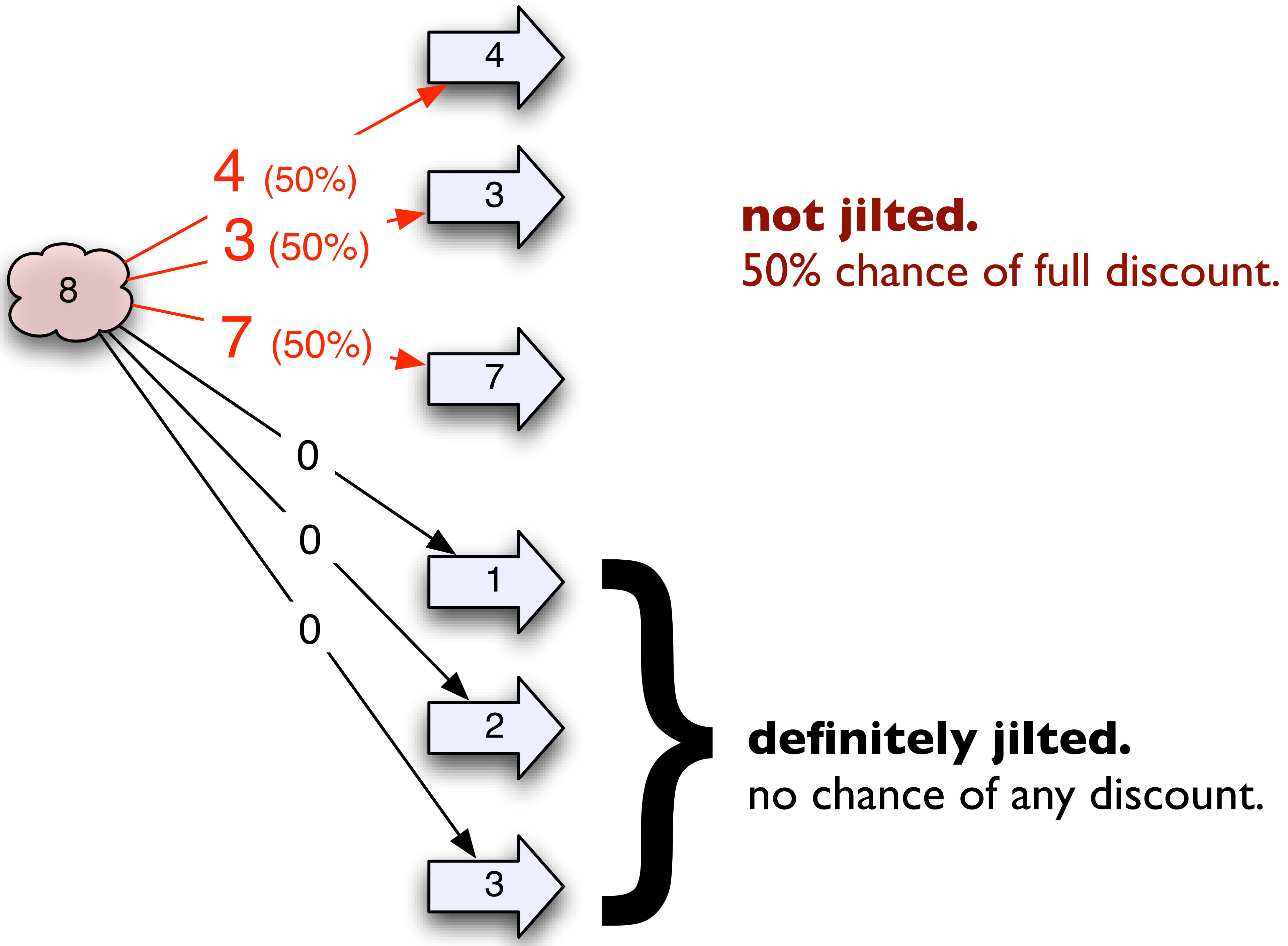
each of girl's boys is either:

- ***jilted*** (girl gives no chance of any discount)
- not jilted (girl gives at least 50% chance of discount)



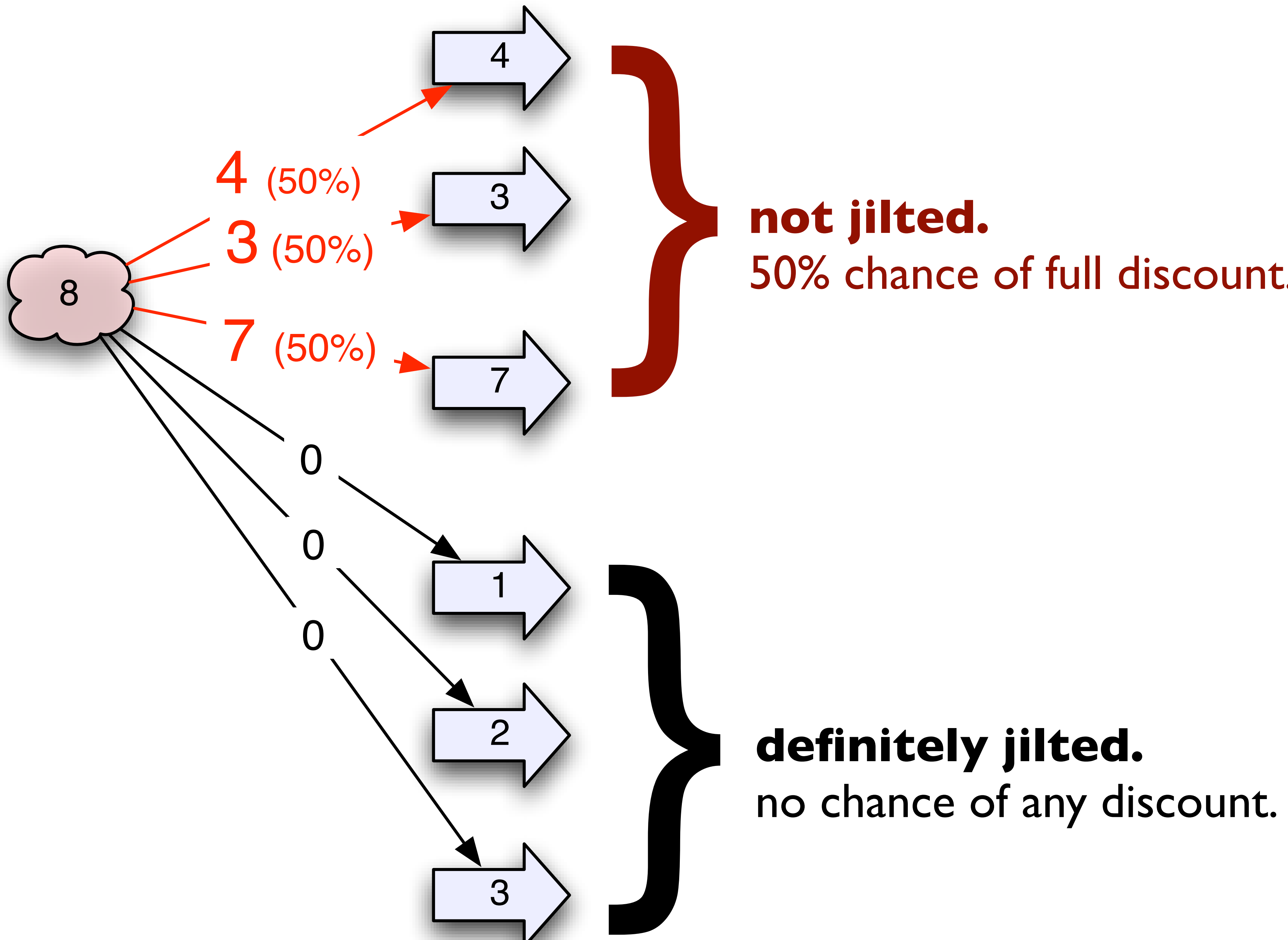
each of girl's boys is either:

- ***jilted*** (girl gives no chance of any discount)
- not jilted (girl gives at least 50% chance of discount)



each of girl's boys is either:

- ***jilted*** (girl gives no chance of any discount)
- not jilted (girl gives at least 50% chance of discount)



recap of algorithm

in each round:

1. Each node randomly chooses to be "boy" or "girl".
   Only edges from boys to higher-cost girls are used.

2. To form stars, each boy chooses a random neighbor (girl).

3. To allocate discounts within stars:

   (a) Each girl allocates greedily in alphabetical order.
   (b) If a boy is partially allocated, with probability 1/2,
   she gives a full discount to just that boy.

analysis

- Guaranteed to return a 2-approximate solution, since it implements the edge-discount algorithm.
- What about running time?

Goal: Show O(log n) rounds (w.h.p.).

analysis of number of rounds

- "Delete" edges when one endpoint's cost becomes zero.

lemma: *In each round, in expectation,*

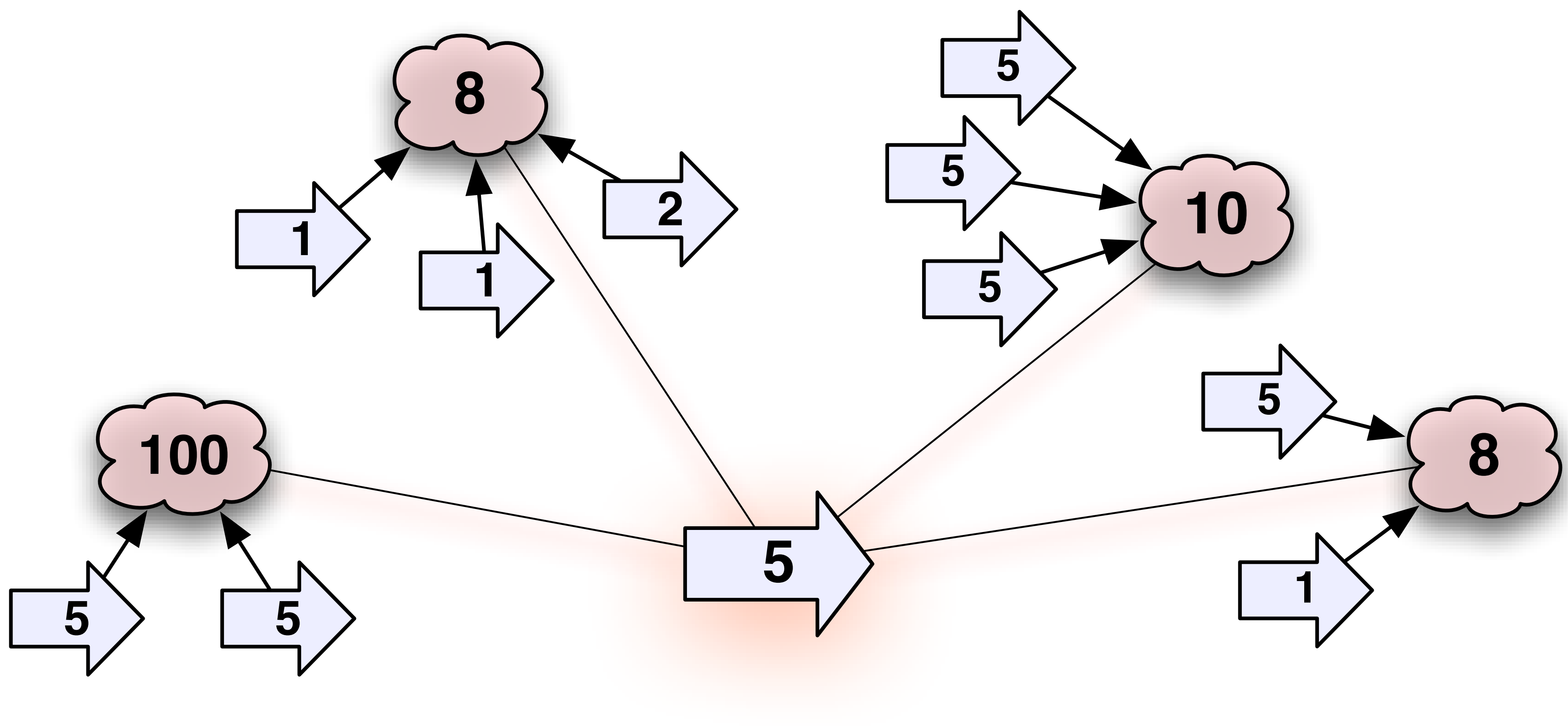
a constant fraction of each boy's active edges are deleted.

proof: (next)

corollary: *Number of rounds is $O(\log n^2) = O(\log n)$*

in expectation and with high probability.

lemma: *In each round, in expectation,*
a constant fraction of ***each boy's*** *active edges are deleted.*

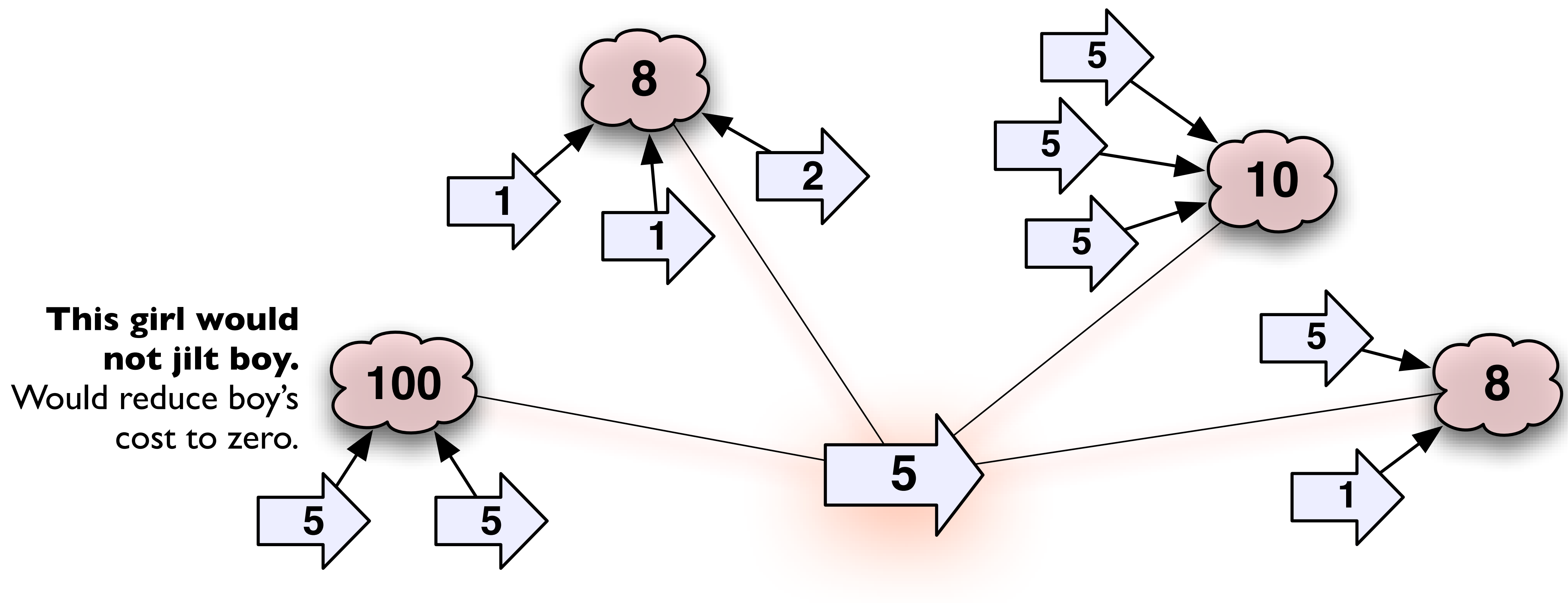


proof: Fix any boy.

For the analysis, condition on the random choices of all other boys.
(Imagine that the boy chooses his girl *after* every other boy chooses.)

- For each girl neighbor, what would happen *if* he were to choose *that* girl?

lemma: *In each round, in expectation, a constant fraction of* ***each boy's*** *active edges are deleted.*

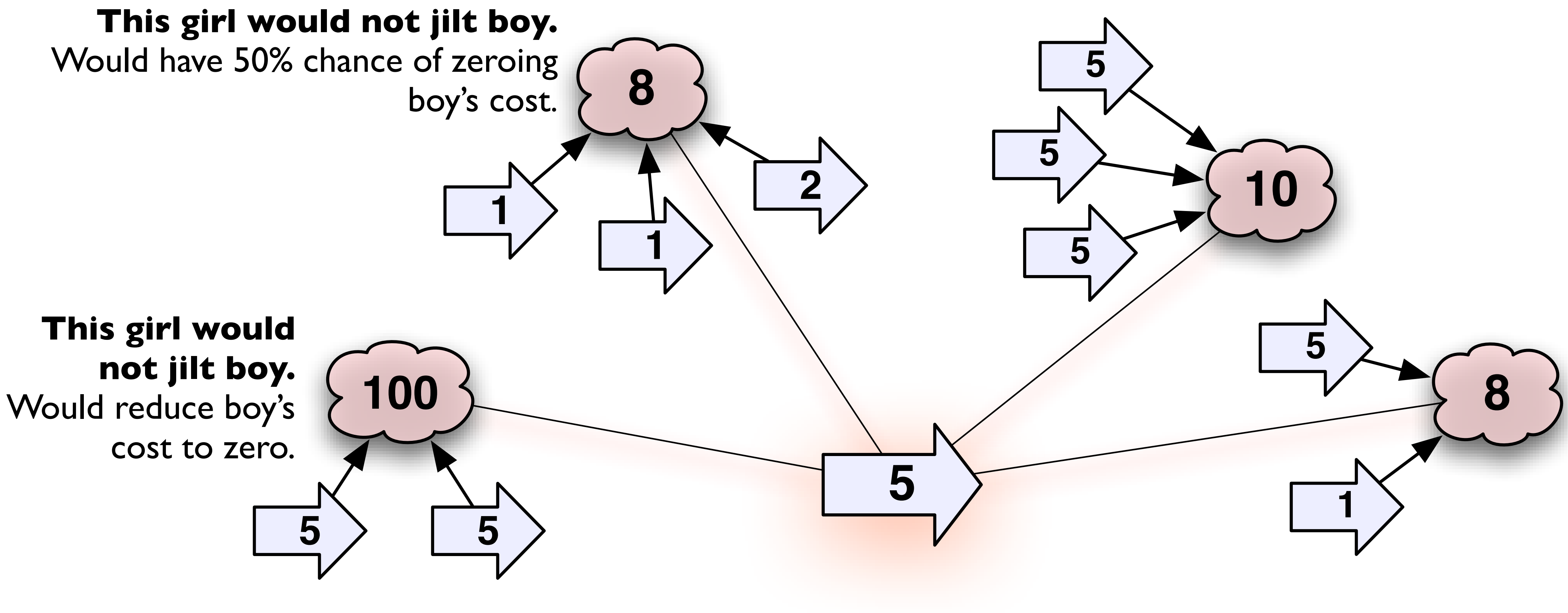


proof: Fix any boy.

For the analysis, condition on the random choices of all other boys.
(Imagine that the boy chooses his girl *after* every other boy chooses.)

- For each girl neighbor, what would happen *if* he were to choose *that* girl?

lemma: *In each round, in expectation,*
a constant fraction of ***each boy's*** *active edges are deleted.*

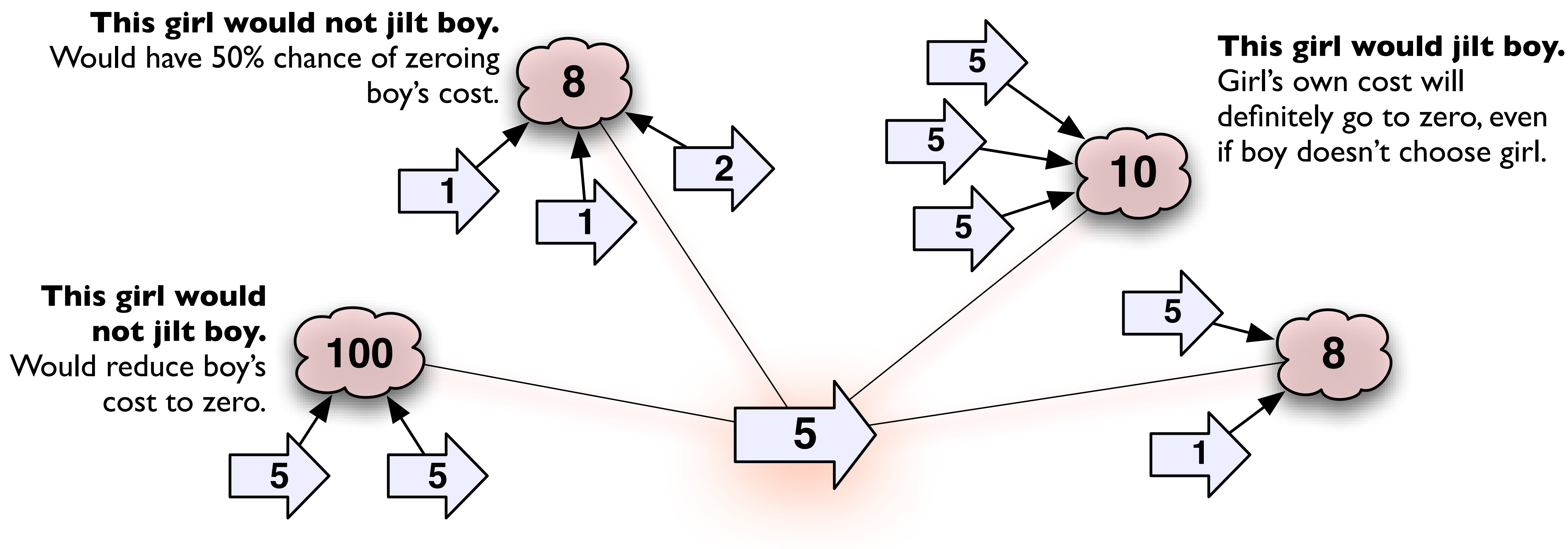


proof: Fix any boy.

For the analysis, condition on the random choices of all other boys.
(Imagine that the boy chooses his girl *after* every other boy chooses.)

- For each girl neighbor, what would happen *if* he were to choose *that* girl?

lemma: *In each round, in expectation, a constant fraction of* ***each boy's*** *active edges are deleted.*

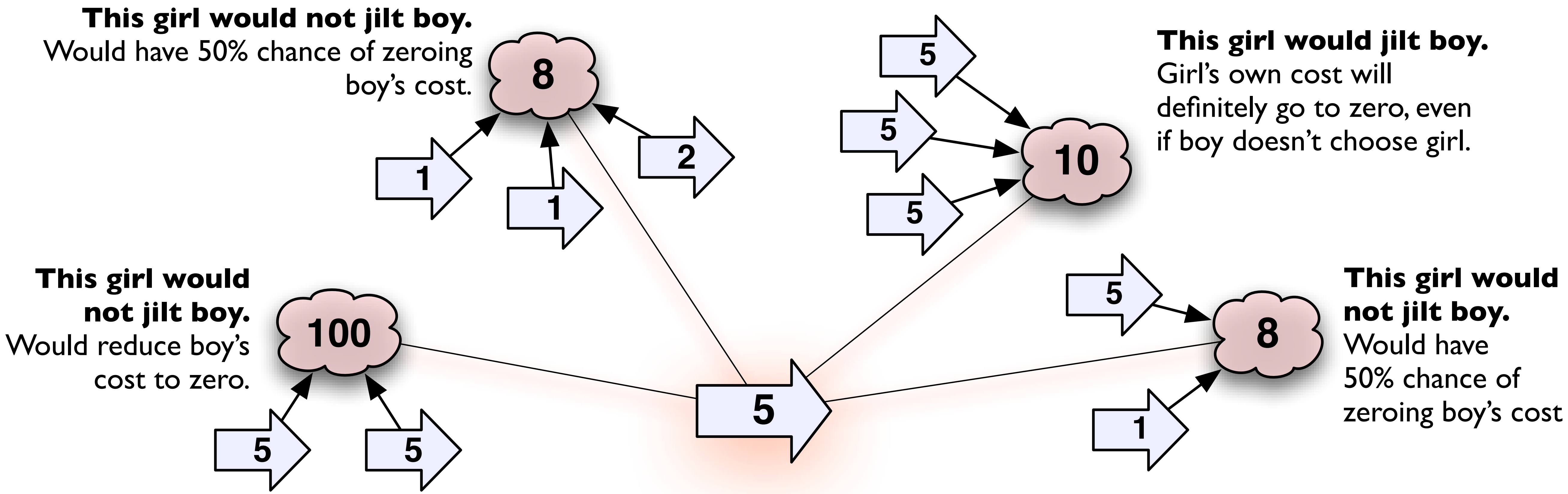


proof: Fix any boy.

For the analysis, condition on the random choices of all other boys.
(Imagine that the boy chooses his girl *after* every other boy chooses.)

- For each girl neighbor, what would happen *if* he were to choose *that* girl?

lemma: *In each round, in expectation, a constant fraction of* ***each boy's*** *active edges are deleted.*

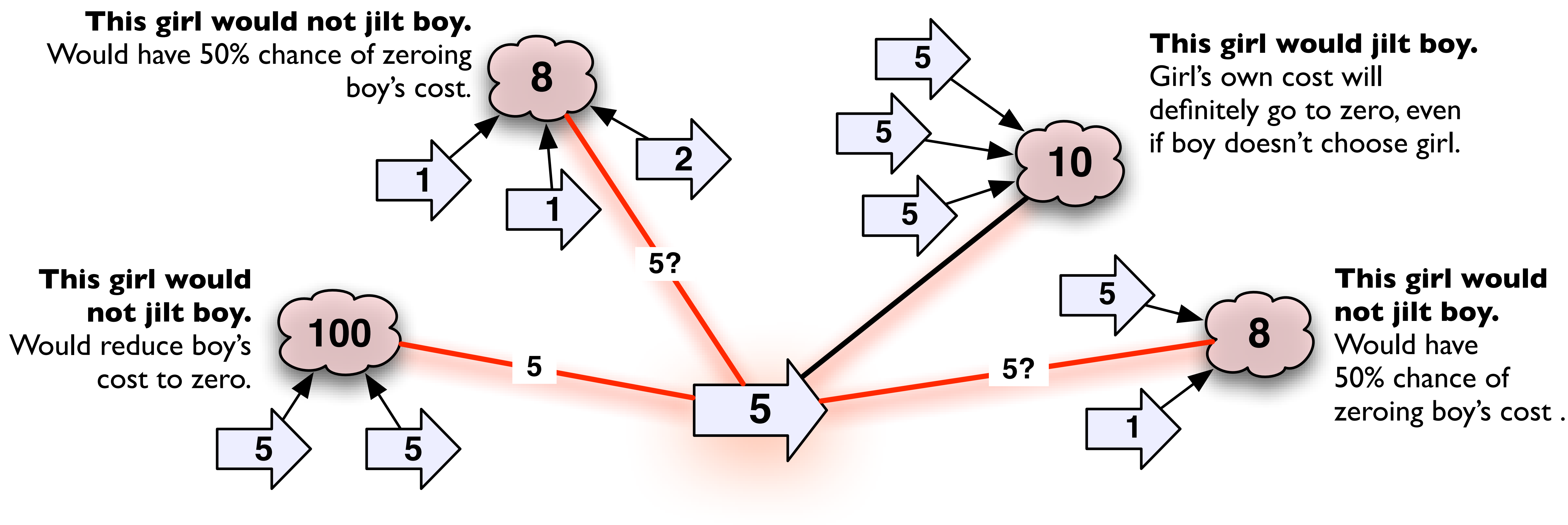


proof: Fix any boy.

For the analysis, condition on the random choices of all other boys.
(Imagine that the boy chooses his girl *after* every other boy chooses.)

- For each girl neighbor, what would happen *if* he were to choose *that* girl?

lemma: *In each round, in expectation,*
a constant fraction of ***each boy's*** *active edges are deleted.*

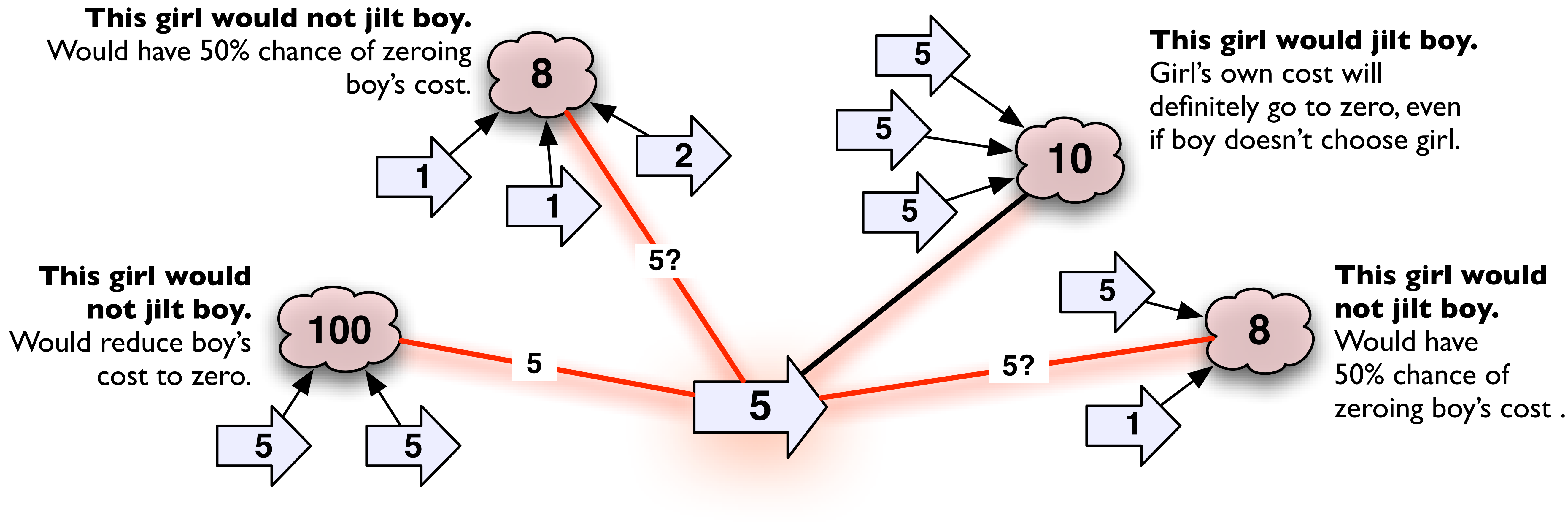


proof: Fix any boy.

For the analysis, condition on the random choices of all other boys.
(Imagine that the boy chooses his girl *after* every other boy chooses.)

- For each girl neighbor, what would happen *if* he were to choose *that* girl?

lemma: *In each round, in expectation, a constant fraction of* ***each boy's*** *active edges are deleted.*



key observation:

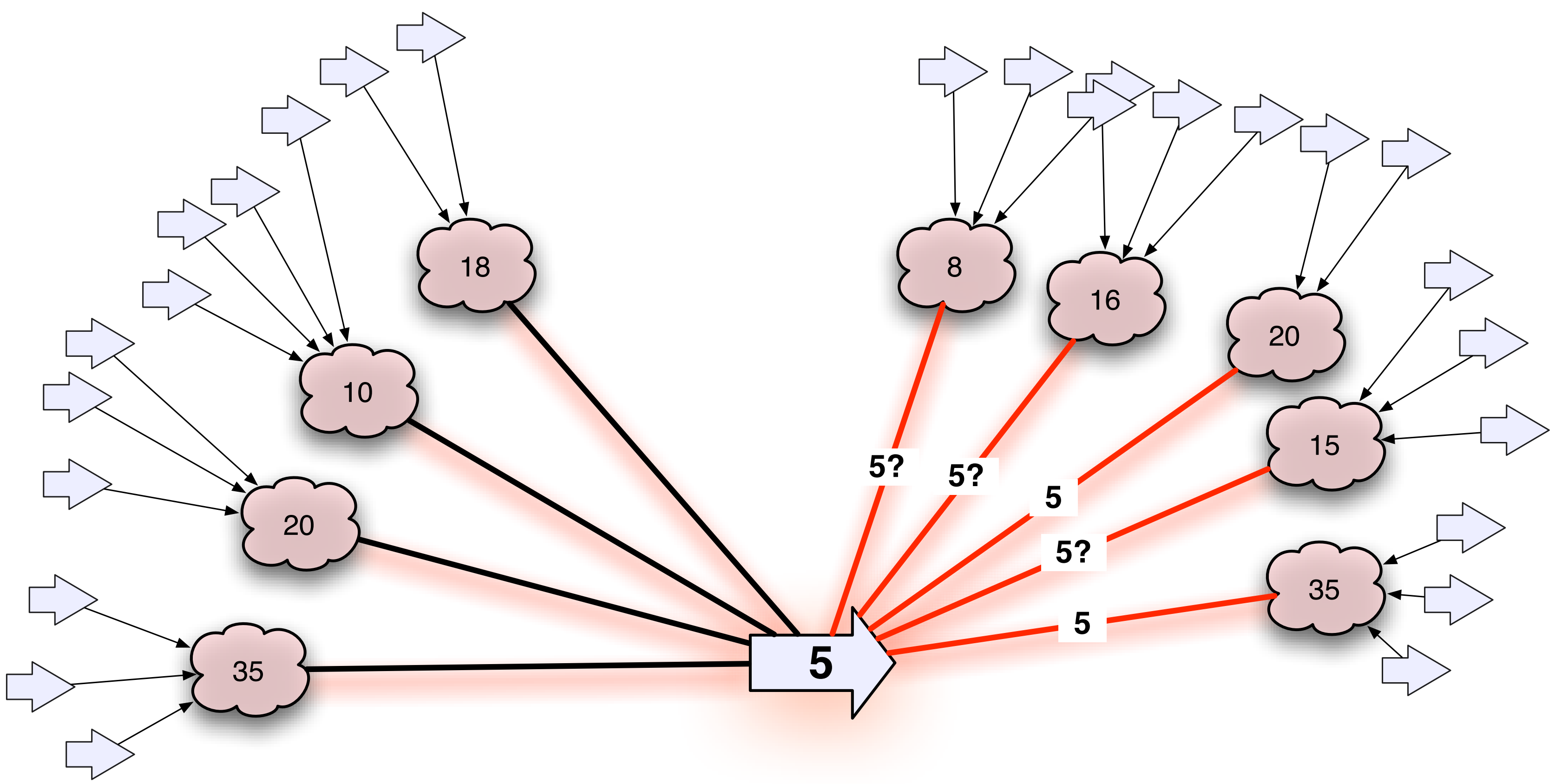
girl **would jilt** boy ⇒ her cost is going to zero regardless of what boy does.

girl **would not jilt** boy ⇒ if boy chooses her, she has at least a 50% chance of zeroing boy's cost...

key observation:

girl **would jilt** boy ⇒ her cost is going to zero regardless of what boy does.

girl **would not jilt** boy ⇒ if boy chooses her,
she has at least a 50% chance of zeroing boy's cost...



case (i): At least half of boy's girls would jilt him.

⇒ At least half of boy's edges will be deleted regardless of what boy does.

case (ii): At least half of boy's girls would not jilt him.

⇒ Boy has at least a 50% chance of choosing a girl
who has at least a 50% chance of zeroing his cost (deleting all his edges).

thank you

- deterministic $O(\log^c n)$-round algorithm?