Blocking Pebbles

M. Fisher, *C. Tennenhouse

University of New England
Biddeford, Maine

CGTC2, 2017

- Graph pebbling

- Graph pebbling
- Two-player pebbling games

- Graph pebbling
- Two-player pebbling games
- BLOCKING PEBBLES

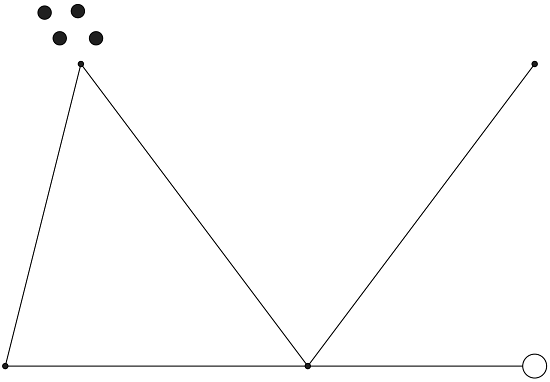
- Graph pebbling
- Two-player pebbling games
- BLOCKING PEBBLES
- Impartial results

- Graph pebbling
- Two-player pebbling games
- BLOCKING PEBBLES
- Impartial results
- Partisan results

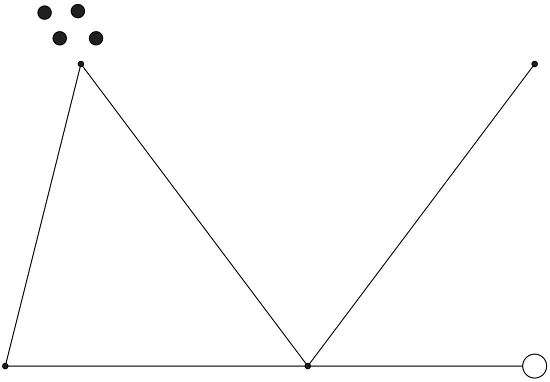
- Graph pebbling
- Two-player pebbling games
- BLOCKING PEBBLES
- Impartial results
- Partisan results
- What's left

Graph pebbling is a one-player game, popular subject among graph theorists. *(Chung, 1989)*

Graph pebbling is a one-player game, popular subject among graph theorists. *(Chung, 1989)*

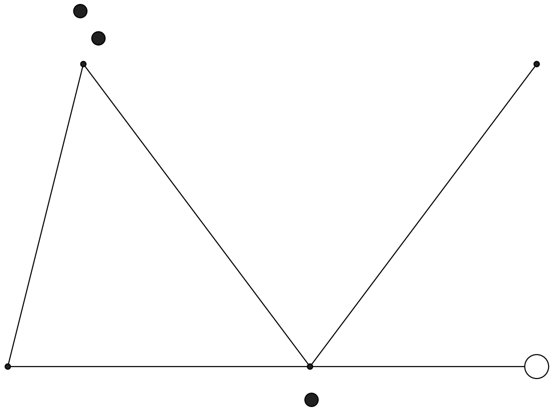


Graph pebbling is a one-player game, popular subject among graph theorists. *(Chung, 1989)*



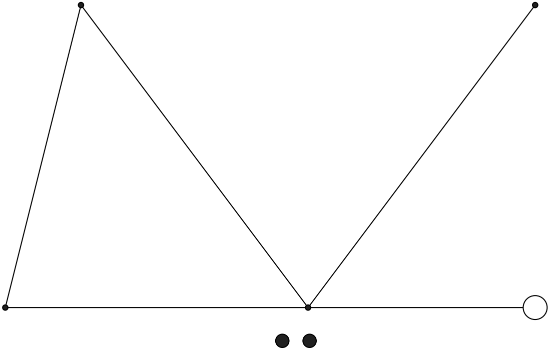
Goal: To get a pebble to the root vertex.

Graph pebbling is a one-player game, popular subject among graph theorists. *(Chung, 1989)*



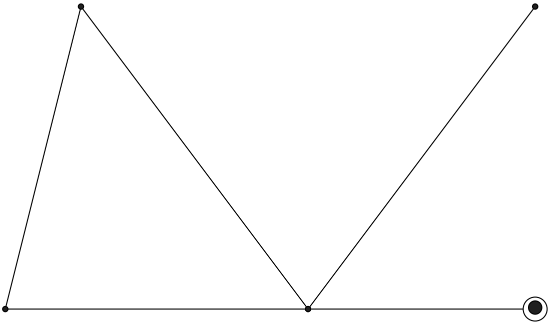
Goal: To get a pebble to the root vertex.

Graph pebbling is a one-player game, popular subject among graph theorists. *(Chung, 1989)*



Goal: To get a pebble to the root vertex.

Graph pebbling is a one-player game, popular subject among graph theorists. *(Chung, 1989)*



Goal: To get a pebble to the root vertex.

$\pi(G, r \in G)$ is the least number of pebbles necessary so that every configuration of π pebbles on G has a strategy to get a pebble to the vertex r.

$\pi(G, r \in G)$ is the least number of pebbles necessary so that every configuration of π pebbles on G has a strategy to get a pebble to the vertex r.

The *pebbling number* $\pi(G) = \min_{r \in G} \pi(G, r)$.

$\pi(K_n) = n$

$\pi(G, r \in G)$ is the least number of pebbles necessary so that every configuration of π pebbles on G has a strategy to get a pebble to the vertex r.

The *pebbling number* $\pi(G) = \min_{r \in G} \pi(G, r)$.

$\pi(K_n) = n$, $\pi(P_n) = 2^{n+1}$

$\pi(G, r \in G)$ is the least number of pebbles necessary so that every configuration of π pebbles on G has a strategy to get a pebble to the vertex r.

The *pebbling number* $\pi(G) = \min_{r \in G} \pi(G, r)$.

$\pi(K_n) = n$, $\pi(P_n) = 2^{n+1}$, $\pi(Q_n) = 2^n$

$\pi(G, r \in G)$ is the least number of pebbles necessary so that every configuration of π pebbles on G has a strategy to get a pebble to the vertex r.

The *pebbling number* $\pi(G) = \min_{r \in G} \pi(G, r)$.

$\pi(K_n) = n$, $\pi(P_n) = 2^{n+1}$, $\pi(Q_n) = 2^n$

Applications in number theory, resource allocation

Modifications to Graph Pebbling:

Modifications to Graph Pebbling:

- Multi-color graph pebbling (Verbeck, 2009)

Modifications to Graph Pebbling:

- Multi-color graph pebbling (Verbeck, 2009)
  - Blocks
  - Two separate colors
  - Not competitive

Modifications to Graph Pebbling:

- Multi-color graph pebbling (Verbeck, 2009)
  - Blocks
  - Two separate colors
  - Not competitive
- Two-player graph pebbling (Prudente, 2015)

Modifications to Graph Pebbling:

- Multi-color graph pebbling (Verbeck, 2009)
  - Blocks
  - Two separate colors
  - Not competitive
- Two-player graph pebbling (Prudente, 2015)
  - Mover vs. Defender
  - Goal is pebbling the root

BLOCKING PEBBLES

BLOCKING PEBBLES

- Directed graph

BLOCKING PEBBLES

- Directed graph
- Red /bLue /Green pebbles

BLOCKING PEBBLES

- Directed graph
- Red /bLue /Green pebbles
- Forward 1 at cost of 1

BLOCKING PEBBLES

- Directed graph
- Red /bLue /Green pebbles
- Forward 1 at cost of 1
- Backward n at no cost

BLOCKING PEBBLES

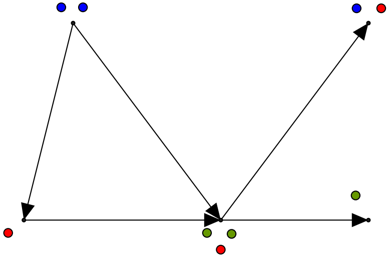
- Directed graph
- Red /bLue /Green pebbles
- Forward 1 at cost of 1
- Backward *n* at no cost
- No joining a red to blue, or a blue to red

Blocking Pebbles

- Directed graph
- Red /bLue /Green pebbles
- Forward 1 at cost of 1
- Backward *n* at no cost
- No joining a red to blue, or a blue to red
- No root

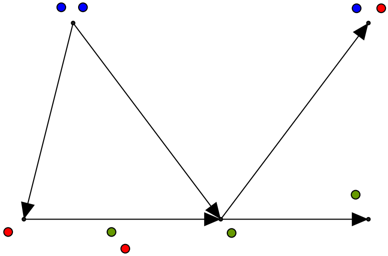
Blocking Pebbles

- Directed graph
- Red /bLue /Green pebbles
- Forward 1 at cost of 1
- Backward *n* at no cost
- No joining a red to blue, or a blue to red
- No root



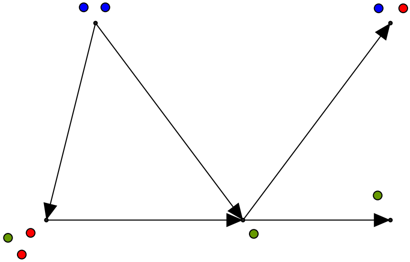
Blocking Pebbles

- Directed graph
- Red /bLue /Green pebbles
- Forward 1 at cost of 1
- Backward *n* at no cost
- No joining a red to blue, or a blue to red
- No root



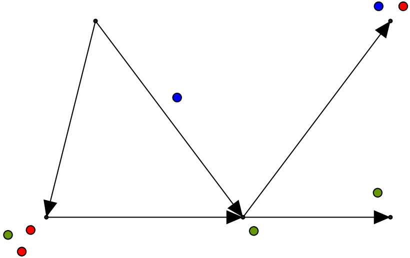
BLOCKING PEBBLES

- Directed graph
- Red /bLue /Green pebbles
- Forward 1 at cost of 1
- Backward *n* at no cost
- No joining a red to blue, or a blue to red
- No root



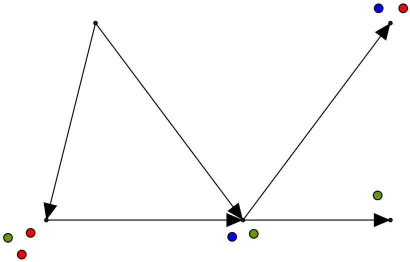
BLOCKING PEBBLES

- Directed graph
- Red /bLue /Green pebbles
- Forward 1 at cost of 1
- Backward *n* at no cost
- No joining a red to blue, or a blue to red
- No root



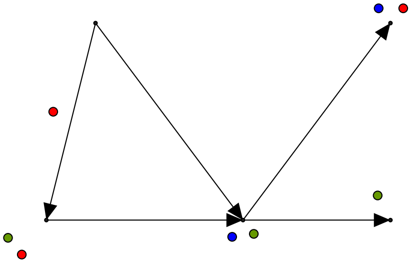
BLOCKING PEBBLES

- Directed graph
- Red /bLue /Green pebbles
- Forward 1 at cost of 1
- Backward *n* at no cost
- No joining a red to blue, or a blue to red
- No root



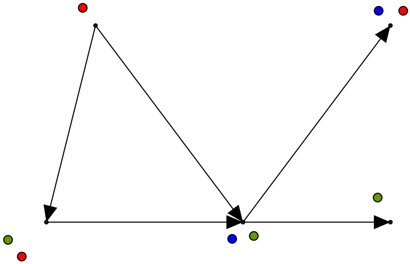
BLOCKING PEBBLES

- Directed graph
- Red /bLue /Green pebbles
- Forward 1 at cost of 1
- Backward *n* at no cost
- No joining a red to blue, or a blue to red
- No root



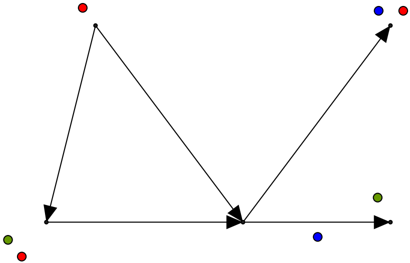
BLOCKING PEBBLES

- Directed graph
- Red /bLue /Green pebbles
- Forward 1 at cost of 1
- Backward *n* at no cost
- No joining a red to blue, or a blue to red
- No root



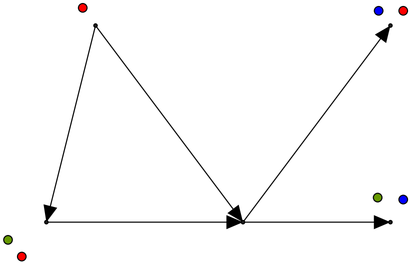
Blocking Pebbles

- Directed graph
- Red /bLue /Green pebbles
- Forward 1 at cost of 1
- Backward n at no cost
- No joining a red to blue, or a blue to red
- No root



BLOCKING PEBBLES

- Directed graph
- Red /bLue /Green pebbles
- Forward 1 at cost of 1
- Backward *n* at no cost
- No joining a red to blue, or a blue to red
- No root



Impartial results:

Impartial results:
With just green pebbles the game is impartial.

Impartial results:
With just green pebbles the game is impartial.

Out-stars

Impartial results:

With just green pebbles the game is impartial.

Out-stars

- Leaf heap can be reduced to leave nim sum of zero

Impartial results:

With just green pebbles the game is impartial.

Out-stars

- Leaf heap can be reduced to leave nim sum of zero
- Any move from center can be returned to center at no cost.

Impartial results:

With just green pebbles the game is impartial.

Out-stars

- Leaf heap can be reduced to leave nim sum of zero
- Any move from center can be returned to center at no cost.
- Just like POKER NIM on leaves with dwindling pool of pebbles

Impartial results:

With just green pebbles the game is impartial.

Out-stars

- Leaf heap can be reduced to leave nim sum of zero
- Any move from center can be returned to center at no cost.
- Just like POKER NIM on leaves with dwindling pool of pebbles

How to examine other trees?

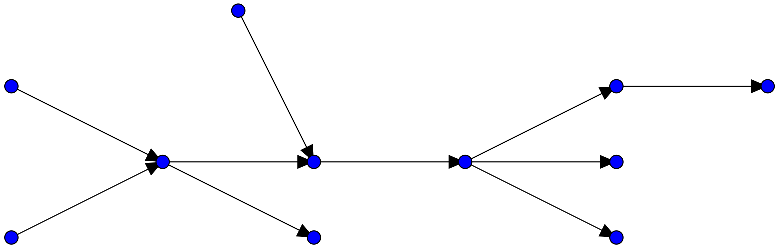
Impartial results:

With just green pebbles the game is impartial.

Out-stars

- Leaf heap can be reduced to leave nim sum of zero
- Any move from center can be returned to center at no cost.
- Just like POKER NIM on leaves with dwindling pool of pebbles

How to examine other trees?



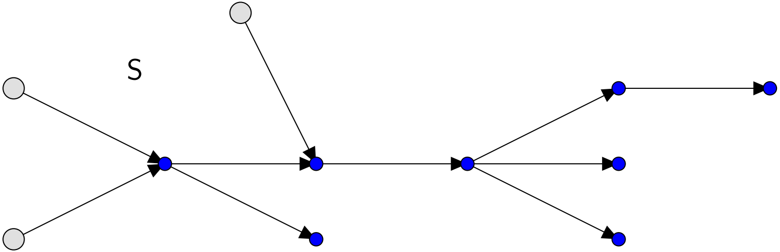
Impartial results:

With just green pebbles the game is impartial.

Out-stars

- Leaf heap can be reduced to leave nim sum of zero
- Any move from center can be returned to center at no cost.
- Just like POKER NIM on leaves with dwindling pool of pebbles

How to examine other trees?



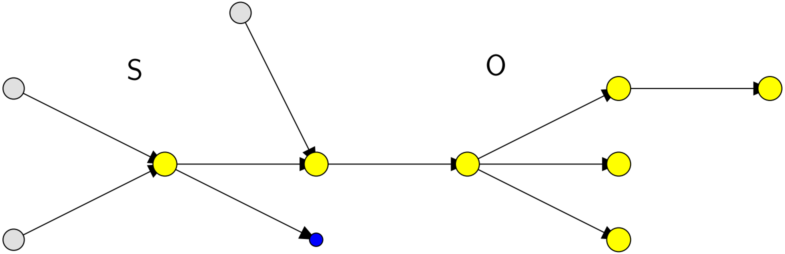
Impartial results:

With just green pebbles the game is impartial.

Out-stars

- Leaf heap can be reduced to leave nim sum of zero
- Any move from center can be returned to center at no cost.
- Just like POKER NIM on leaves with dwindling pool of pebbles

How to examine other trees?



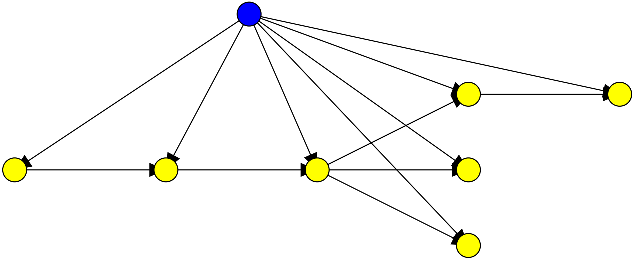
Impartial results:

With just green pebbles the game is impartial.

Out-stars

- Leaf heap can be reduced to leave nim sum of zero
- Any move from center can be returned to center at no cost.
- Just like POKER NIM on leaves with dwindling pool of pebbles

How to examine other trees?



Note that this reduction turns an in-star into a single edge, so the value is $*c$.

Note that this reduction turns an in-star into a single edge, so the value is $*c$.

Theorem

The path game $(p_1, p_2, \ldots, p_n)$ *has value* $*(\sum_{\oplus} p_{2i})$

Proof.

Consider a counter-example with lowest sum

Note that this reduction turns an in-star into a single edge, so the value is $*c$.

Theorem

The path game $(p_1, p_2, \ldots, p_n)$ *has value* $*(\sum_{\oplus} p_{2i})$

Proof.

Consider a counter-example with lowest sum then last lexicographically,

Note that this reduction turns an in-star into a single edge, so the value is $*c$.

Theorem

The path game $(p_1, p_2, \ldots, p_n)$ *has value* $*(\sum_{\oplus} p_{2i})$

Proof.

Consider a counter-example with lowest sum then last lexicographically, consider all options, . . . □

Note that this reduction turns an in-star into a single edge, so the value is $*c$.

Theorem

The path game $(p_1, p_2, \ldots, p_n)$ *has value* $*(\sum_{\oplus} p_{2i})$

Proof.

Consider a counter-example with lowest sum then last lexicographically, consider all options, . . . □

Proof.

Reduce the path to a star. □

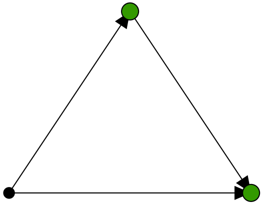
Reduction does not resolve all issues.

Reduction does not resolve all issues.

Some trees reduce to digraphs with weak circuits.

Reduction does not resolve all issues.

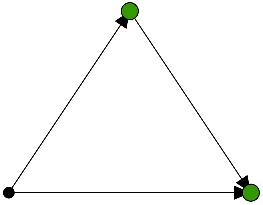
Some trees reduce to digraphs with weak circuits.



Game on Transitive Triple is in $\mathcal{P}$ iff non-source heaps are equal.

Reduction does not resolve all issues.

Some trees reduce to digraphs with weak circuits.



Game on Transitive Triple is in $\mathcal{P}$ iff non-source heaps are equal.

Value of G?

Partisan results:

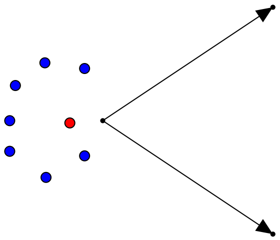
Partisan results:

Proofs are case analyses.

Partisan results:

Proofs are case analyses.

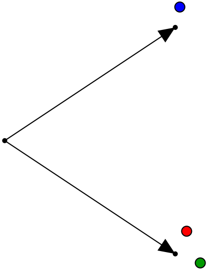
- $n \in \mathbb{Z}$
-
-
-
-



Partisan results:

Proofs are case analyses.

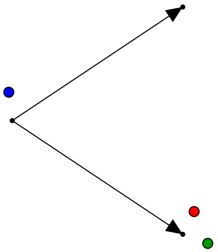
- $n \in \mathbb{Z}$
- $\downarrow$
-
-
-



Partisan results:

Proofs are case analyses.

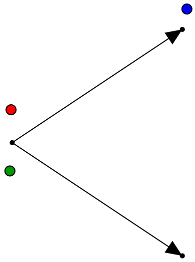
- $n \in \mathbb{Z}$
- $\downarrow$
- $*, 0$
-
-



Partisan results:

Proofs are case analyses.

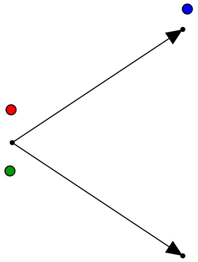
- $n \in \mathbb{Z}$
- $\downarrow$
- $*, 0$
-
-



Partisan results:

Proofs are case analyses.

- $n \in \mathbb{Z}$
- $\downarrow$
- $*, 0$
- Switches
- Lots of stars, paths



Remaining questions

Remaining questions

- A number of diadic rationals we can't get with our code. What's possible?

Remaining questions

- A number of diadic rationals we can't get with our code. What's possible?
- General result for green game? Can we further reduce non-trees?

Remaining questions

- A number of diadic rationals we can't get with our code. What's possible?
- General result for green game? Can we further reduce non-trees?

Thank you!