Transforming Games

COL 2.

Game on a graph.

CoL 2.

Game on a graph.

Red v/s Blue

CoL

Game on a graph.

Red v/s Blue

On your turn pick an uncolored vertex

& color it w/ your color.

CoL 2.

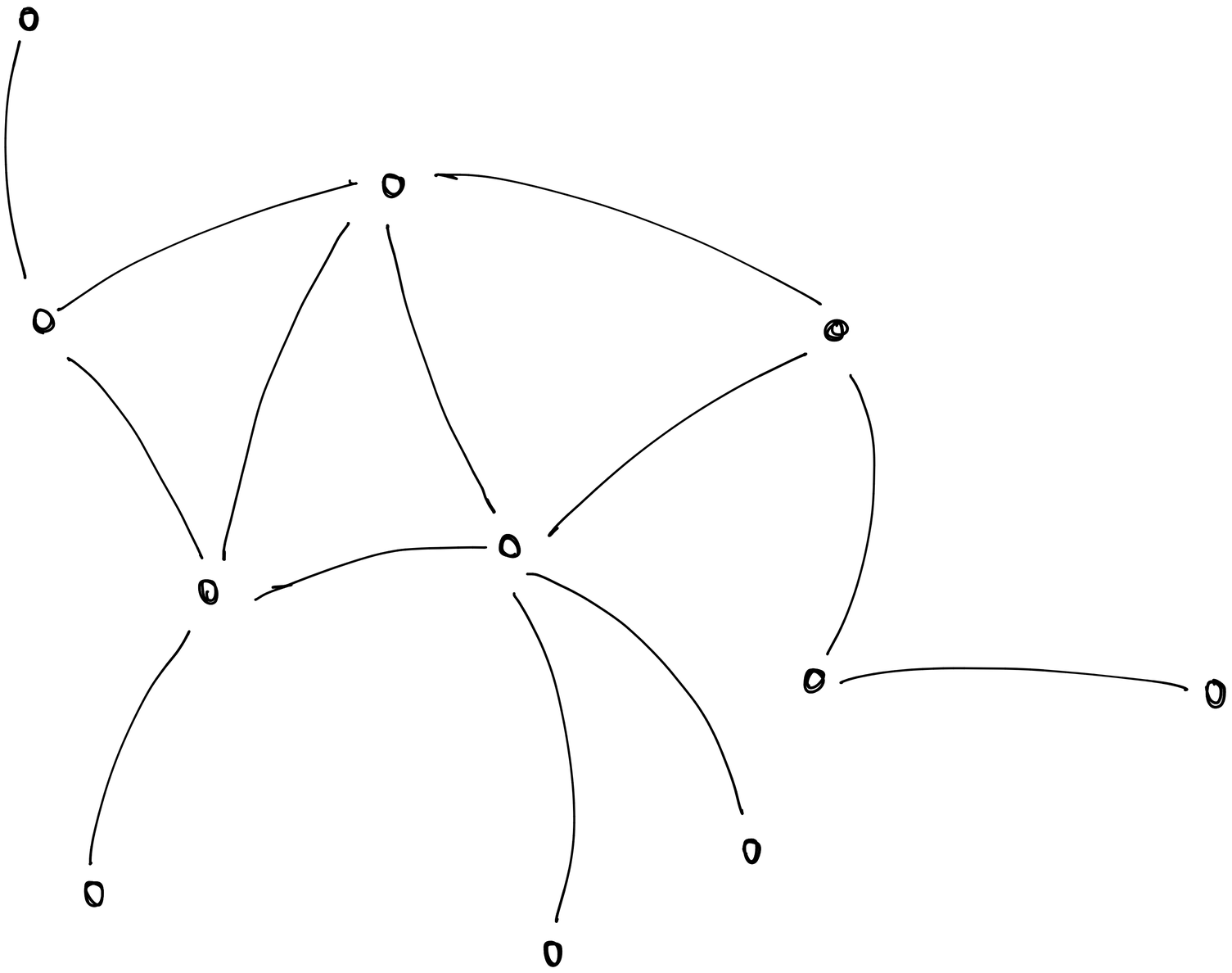
Game on a graph.

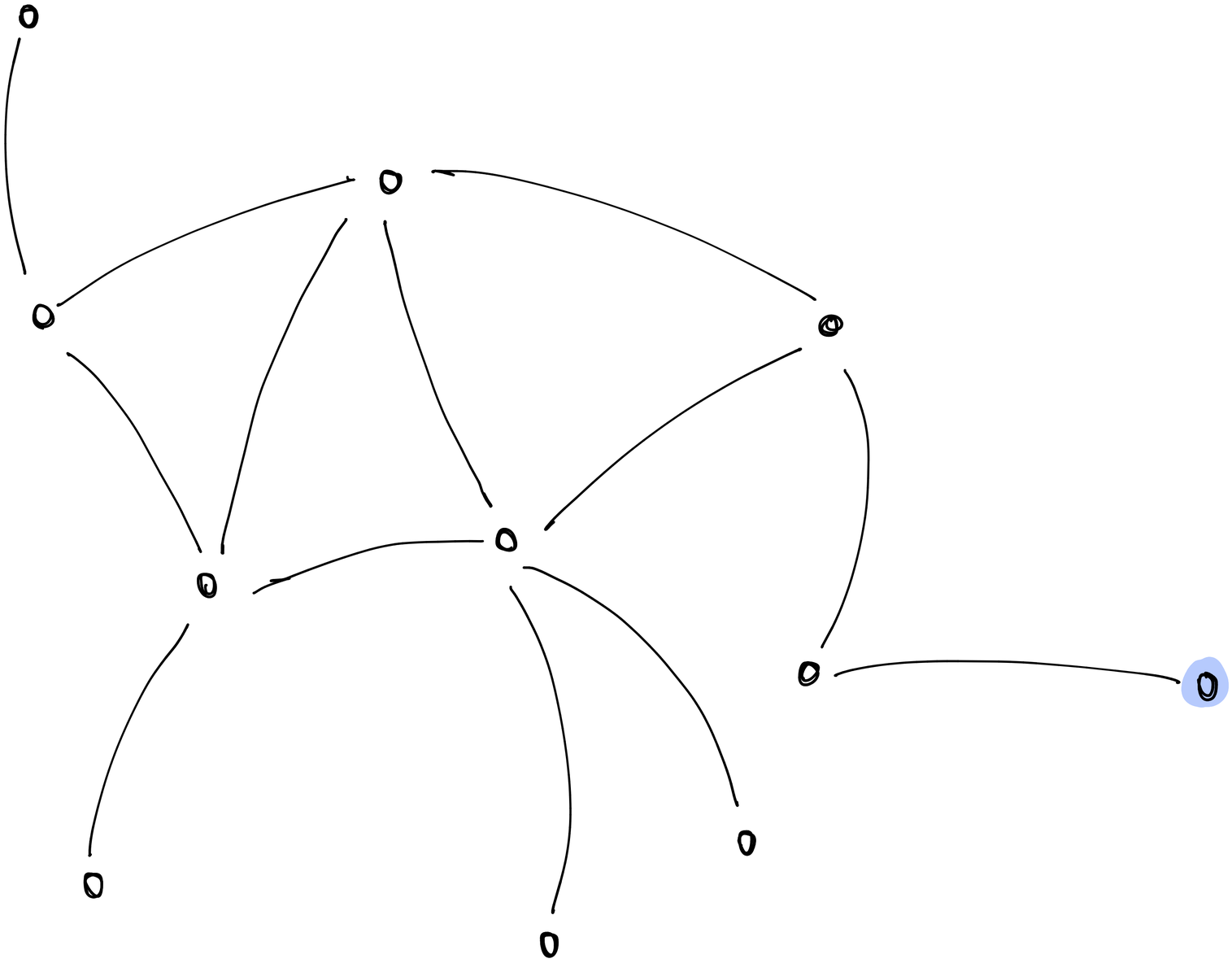
Red v/s Blue

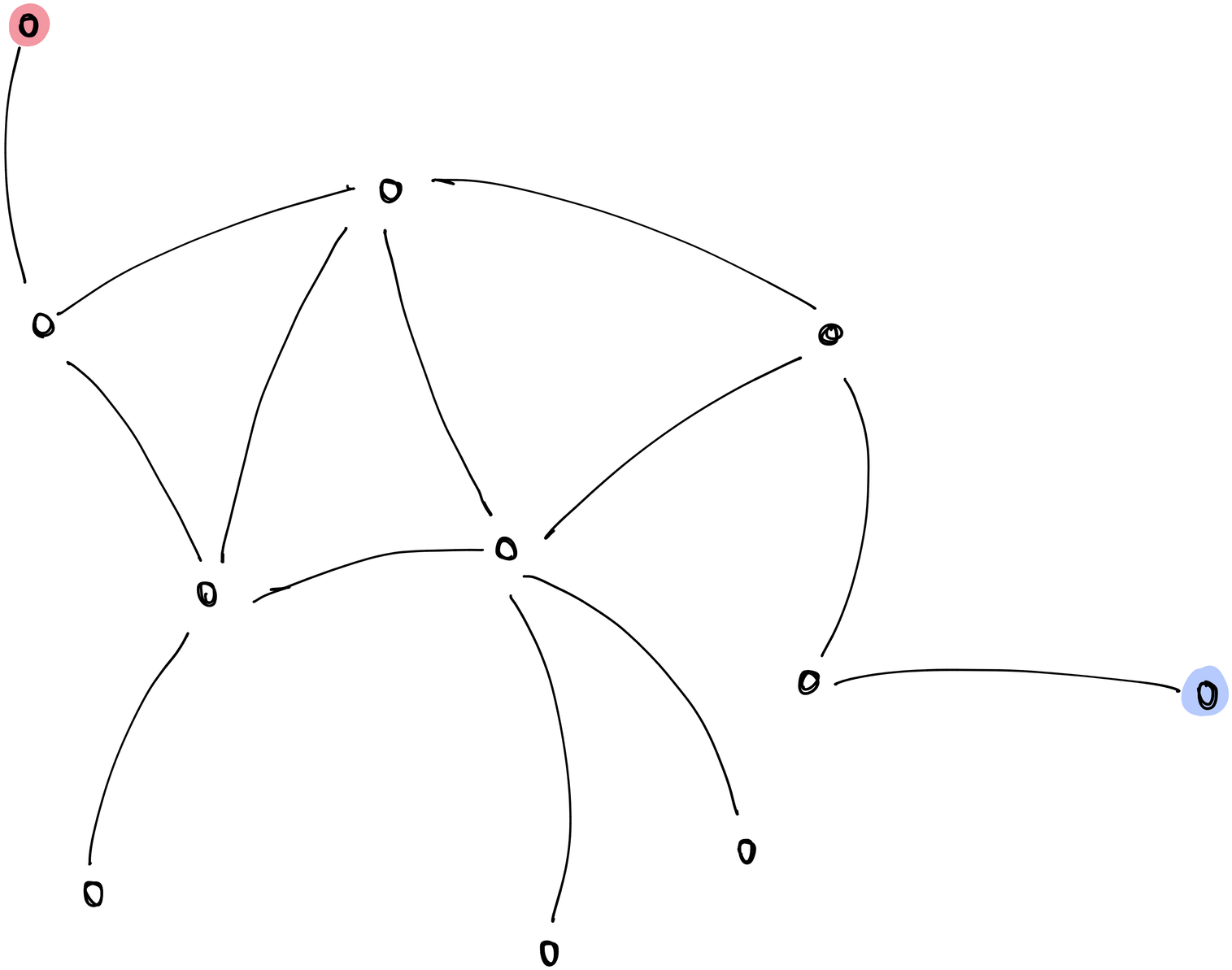
On your turn pick an uncolored vertex

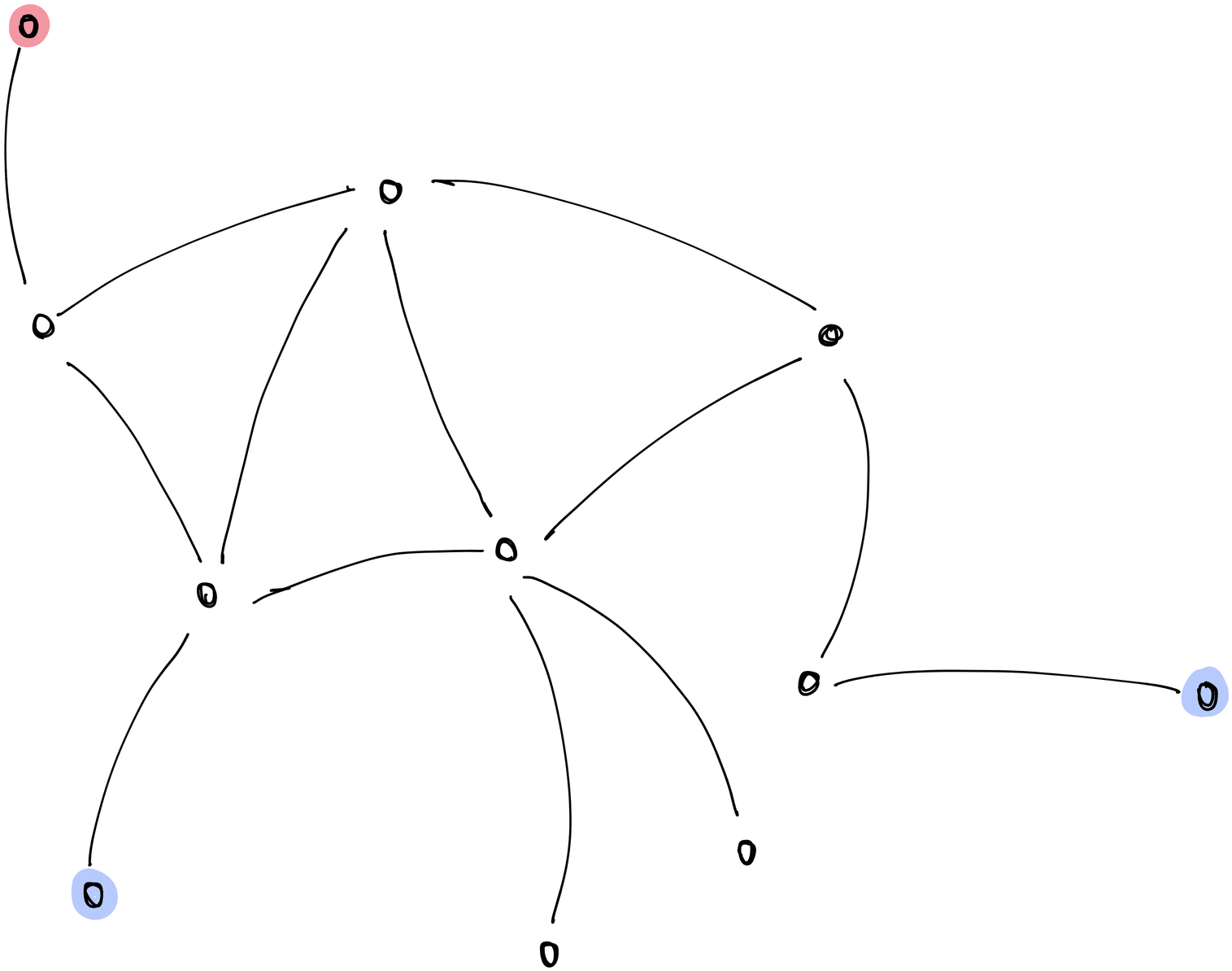
& color it w/ your color.

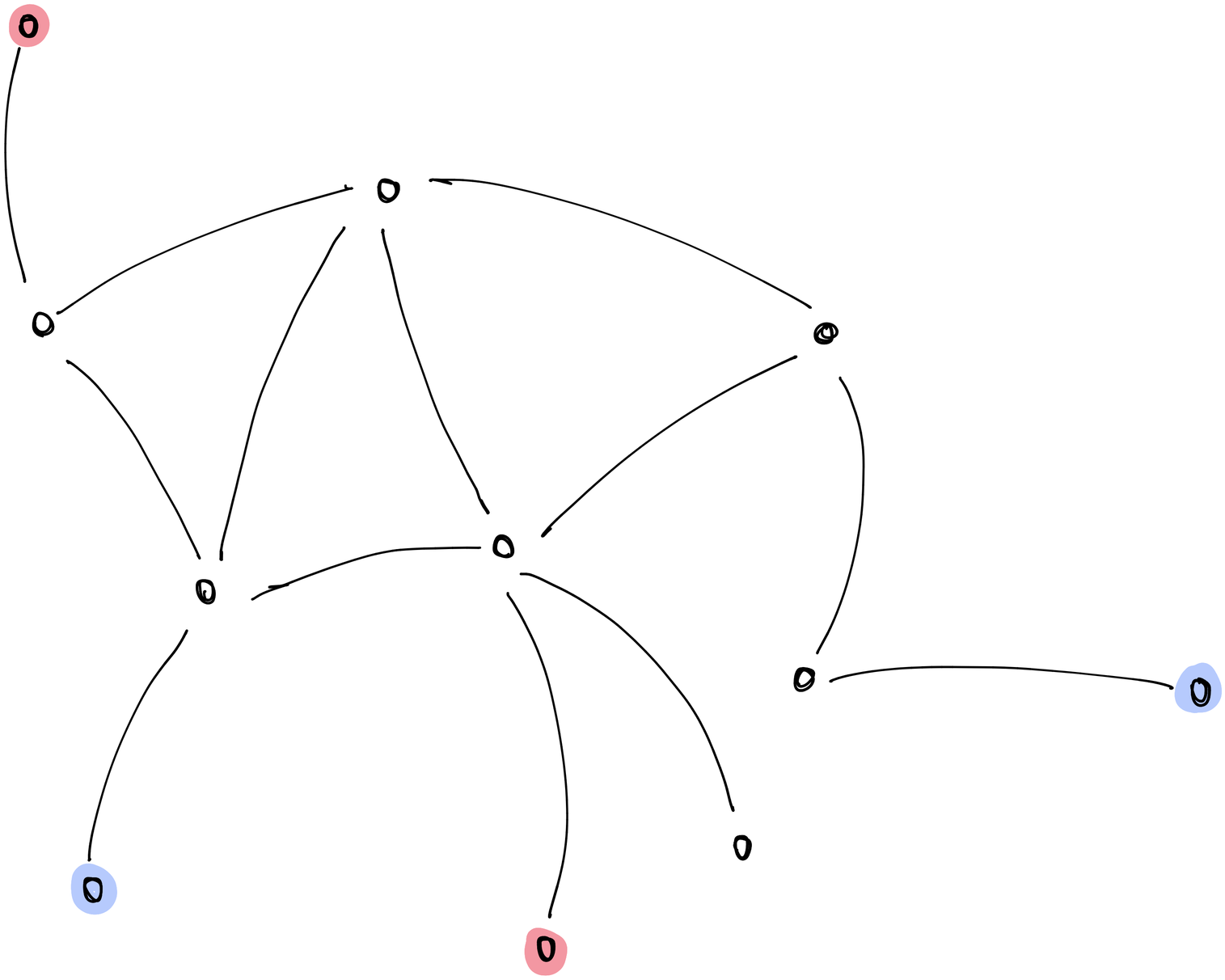
CANNOT be adjacent to a vertex already colored w/ your color.

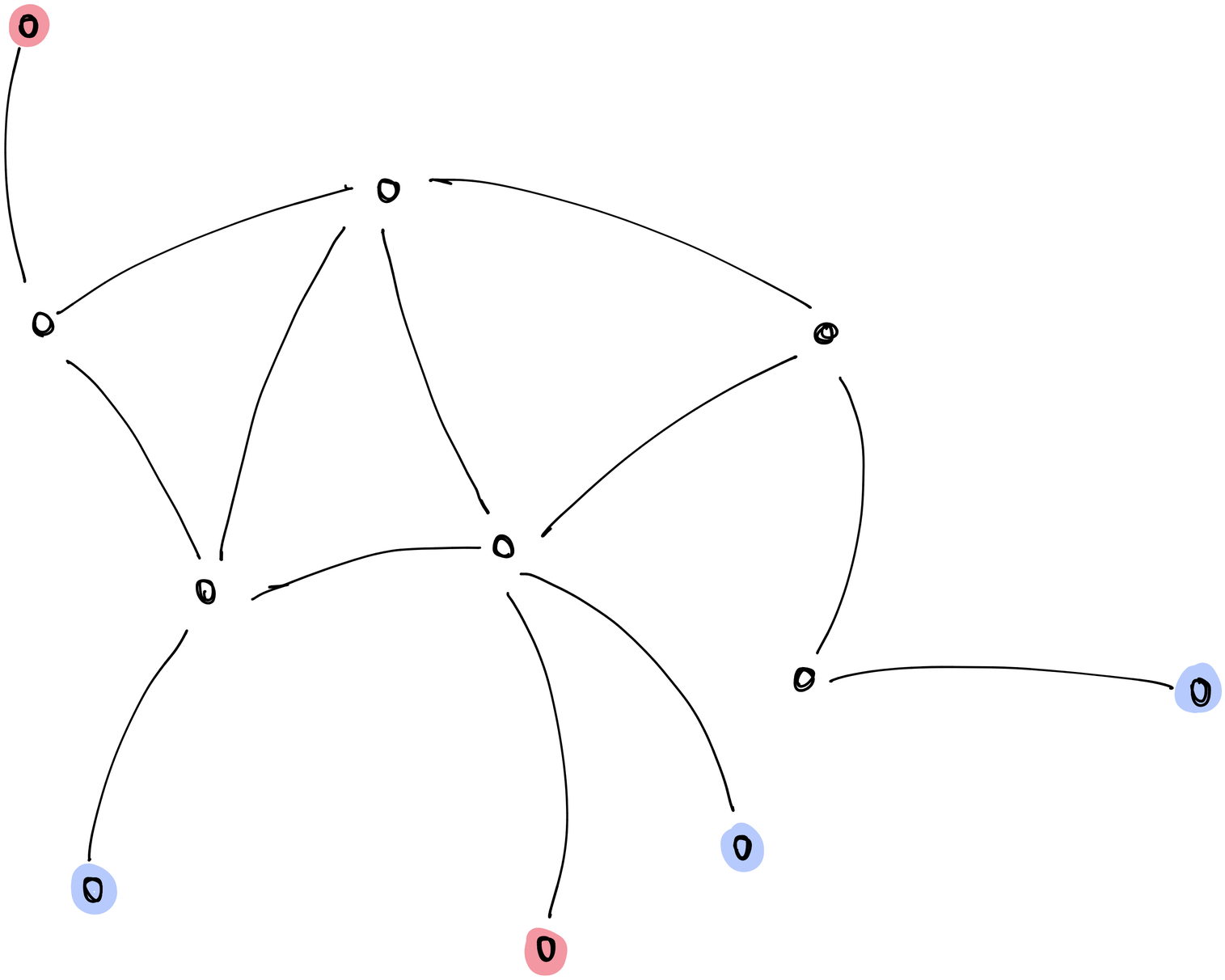


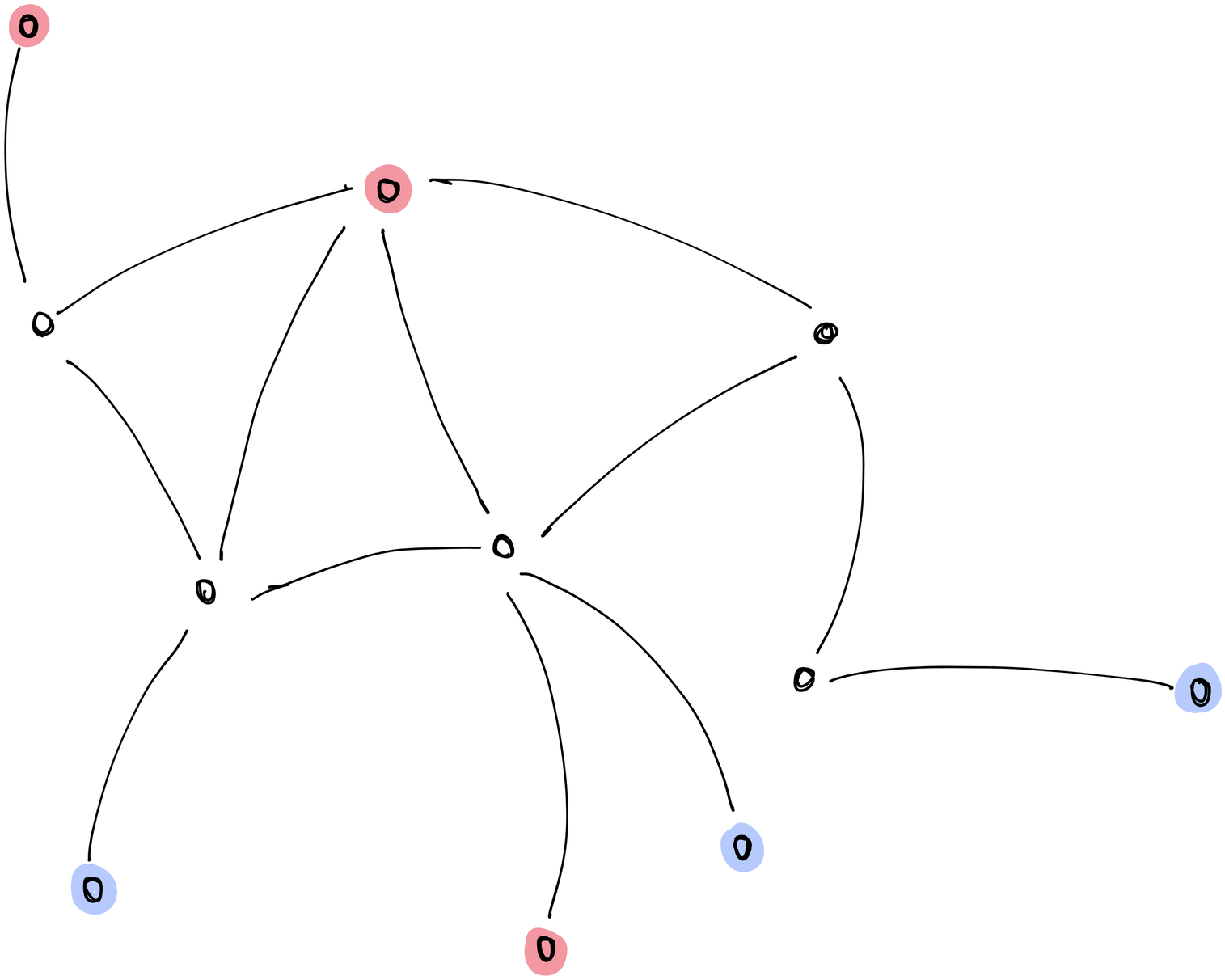


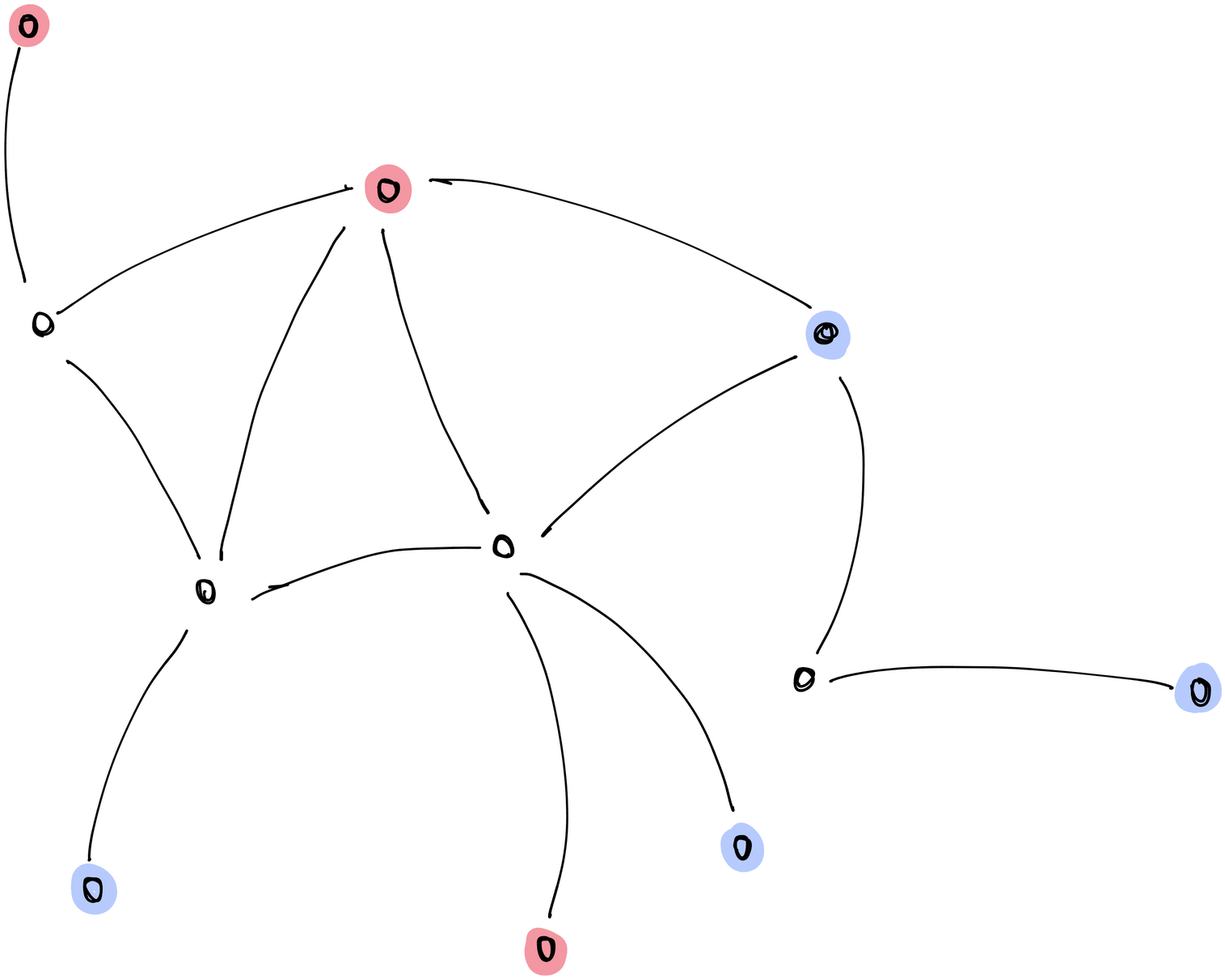


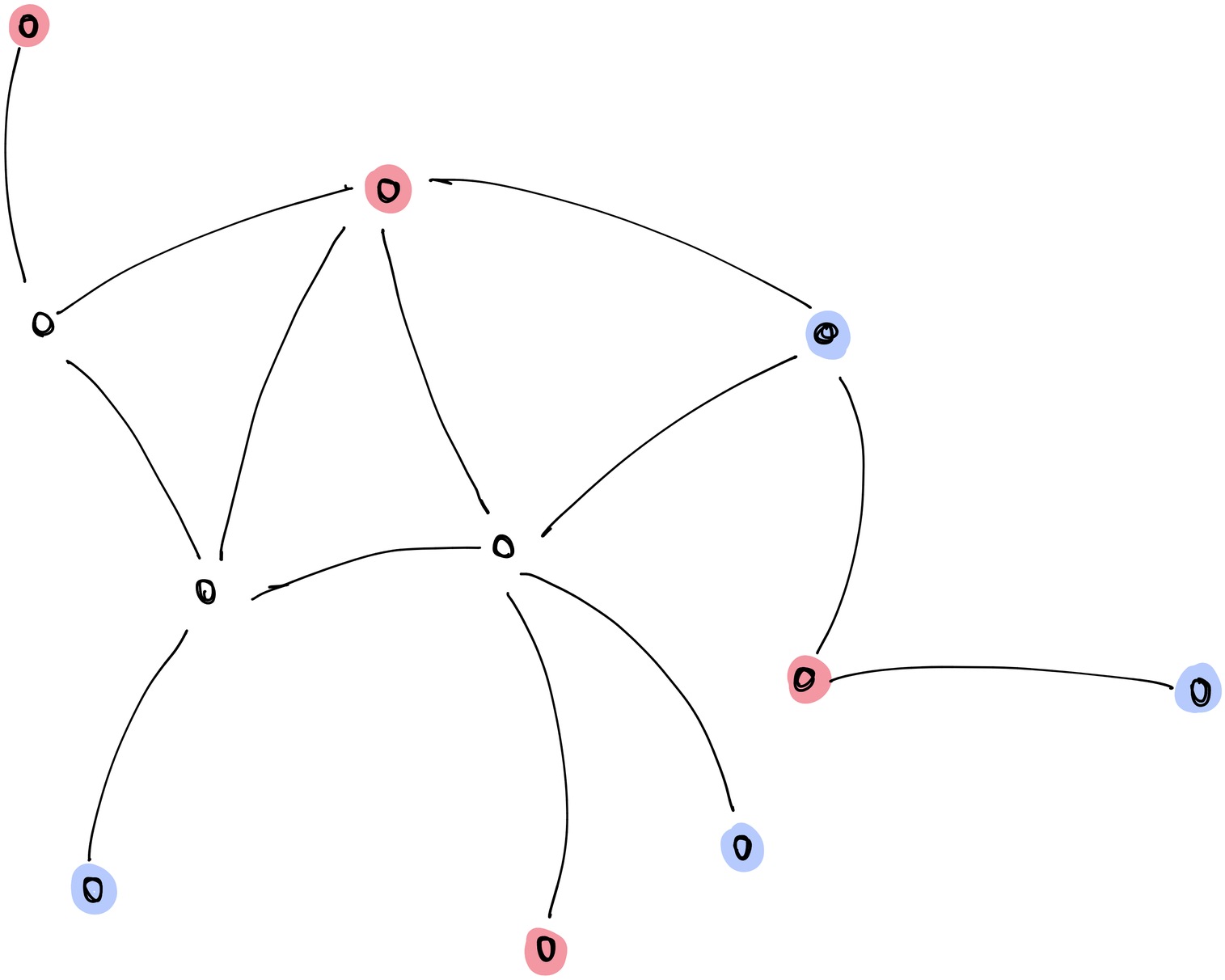


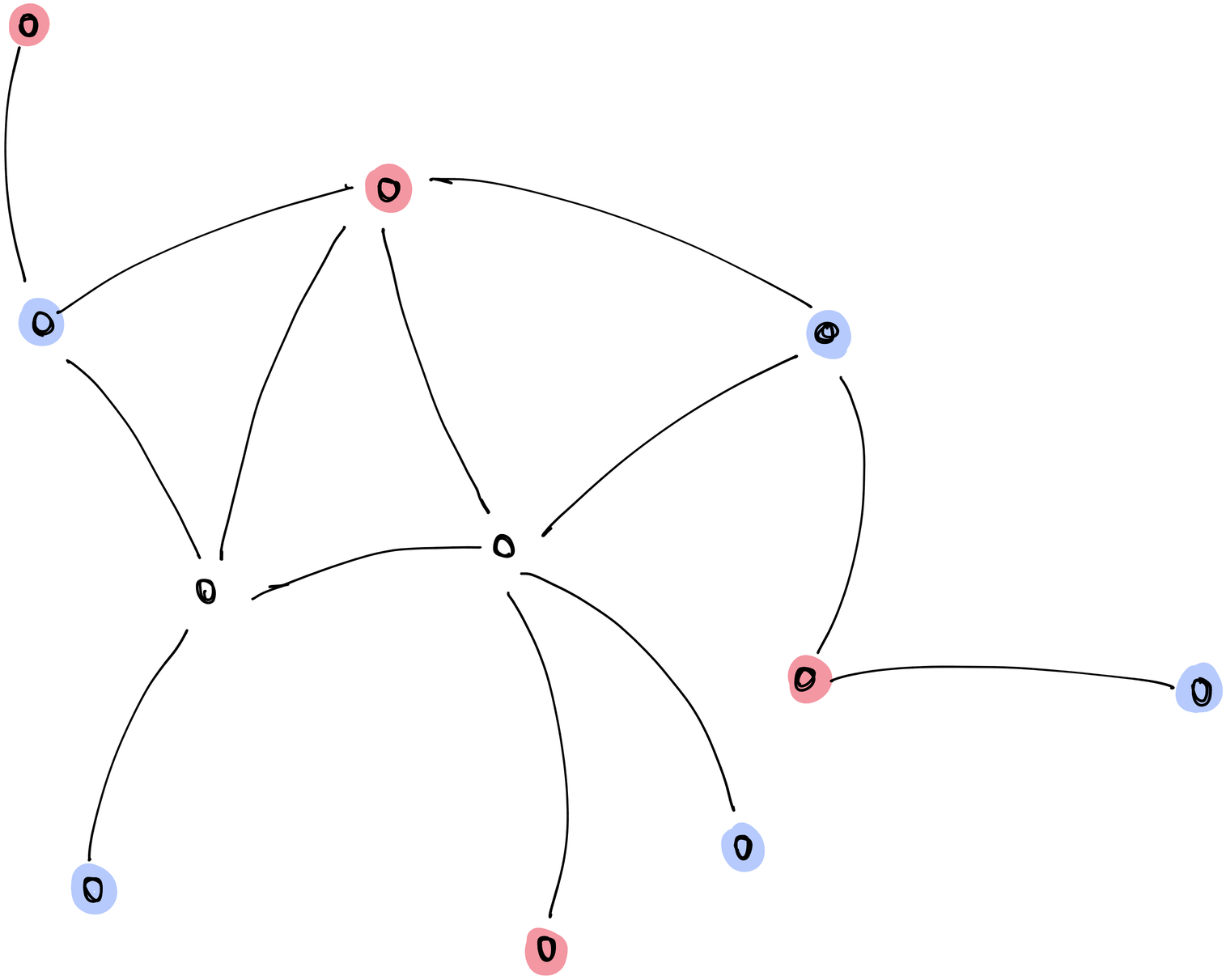




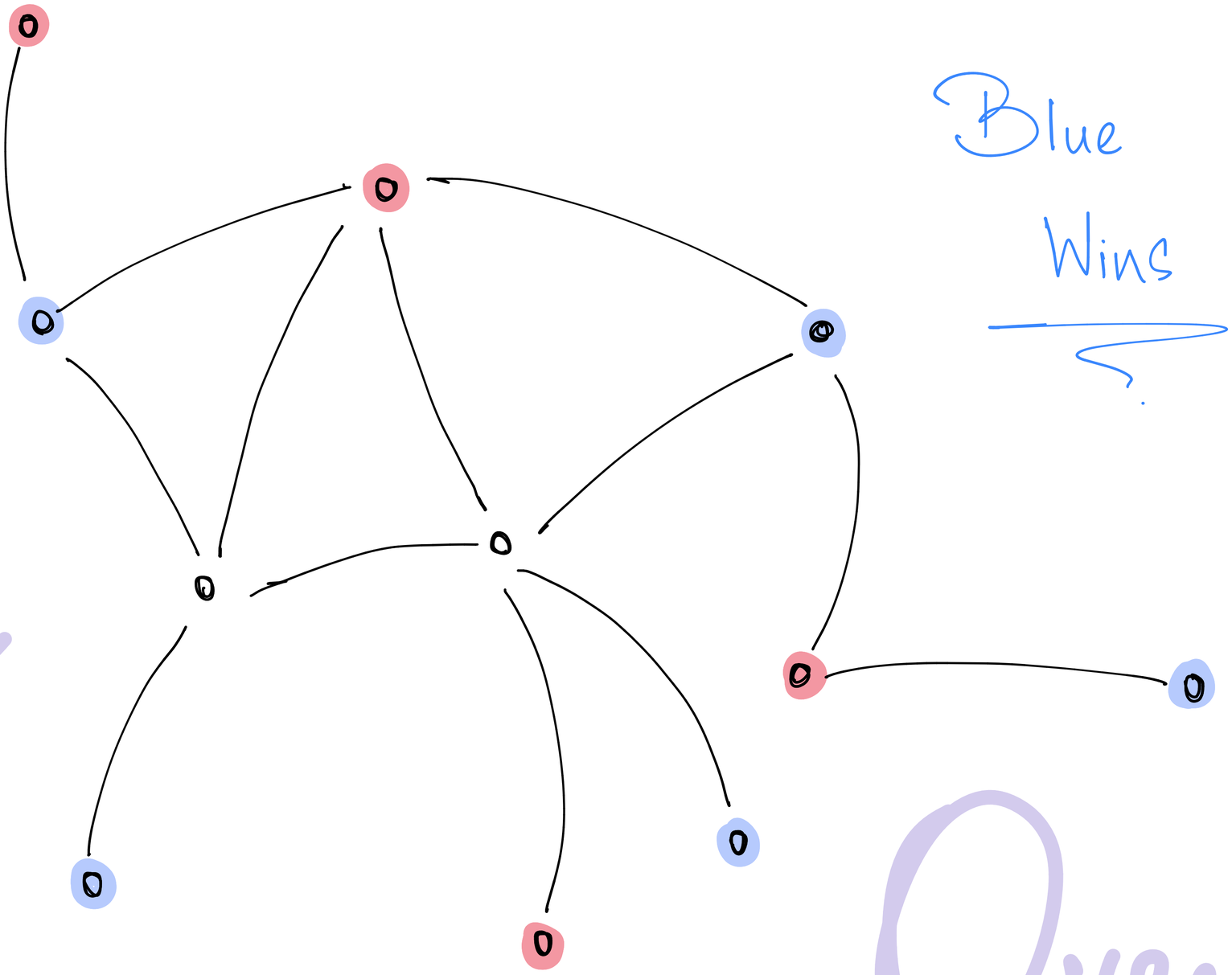








Game



Blue Wins

Over

No Go

Game on a graph.

No Go

Game on a graph.

Red v/s Blue

No Go

Game on a graph.

Red v/s Blue

On your turn pick an uncolored vertex

& color it w/ your color.

No Go

Game on a graph.

Red v/s Blue

On your turn pick an uncolored vertex

& color it w/ your color.

No Go

Game on a graph.

Red v/s Blue

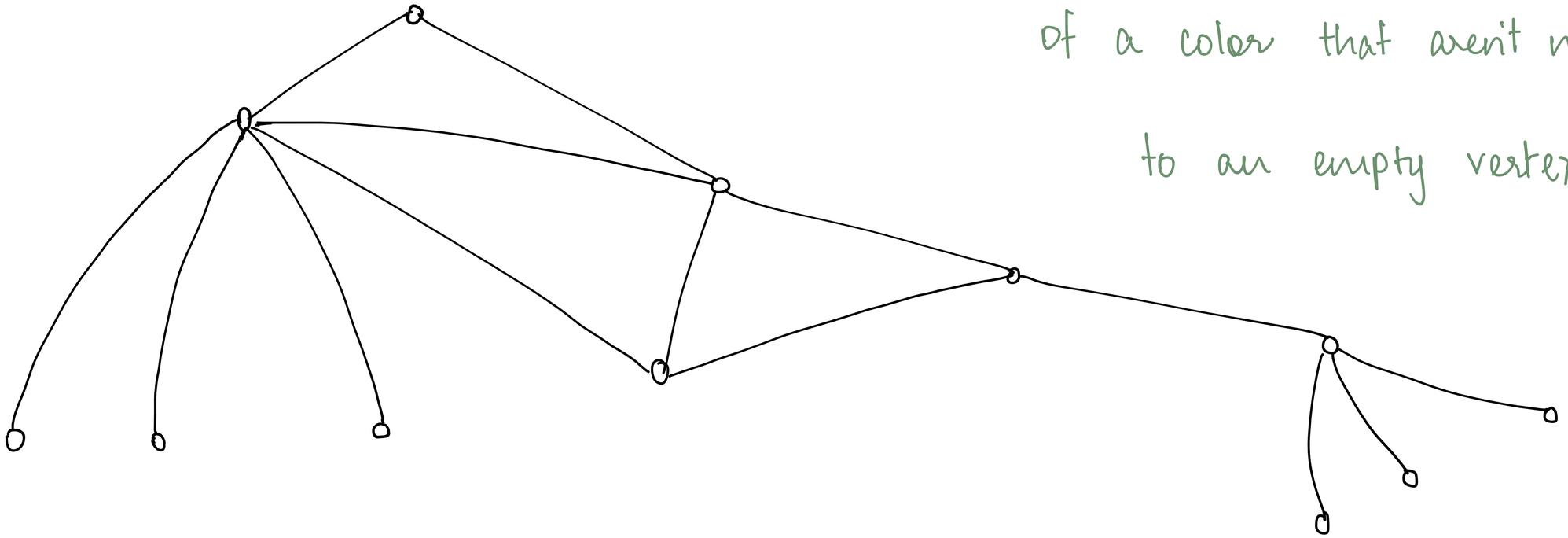
On your turn pick an uncolored vertex

& color it w/ your color.

* You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.

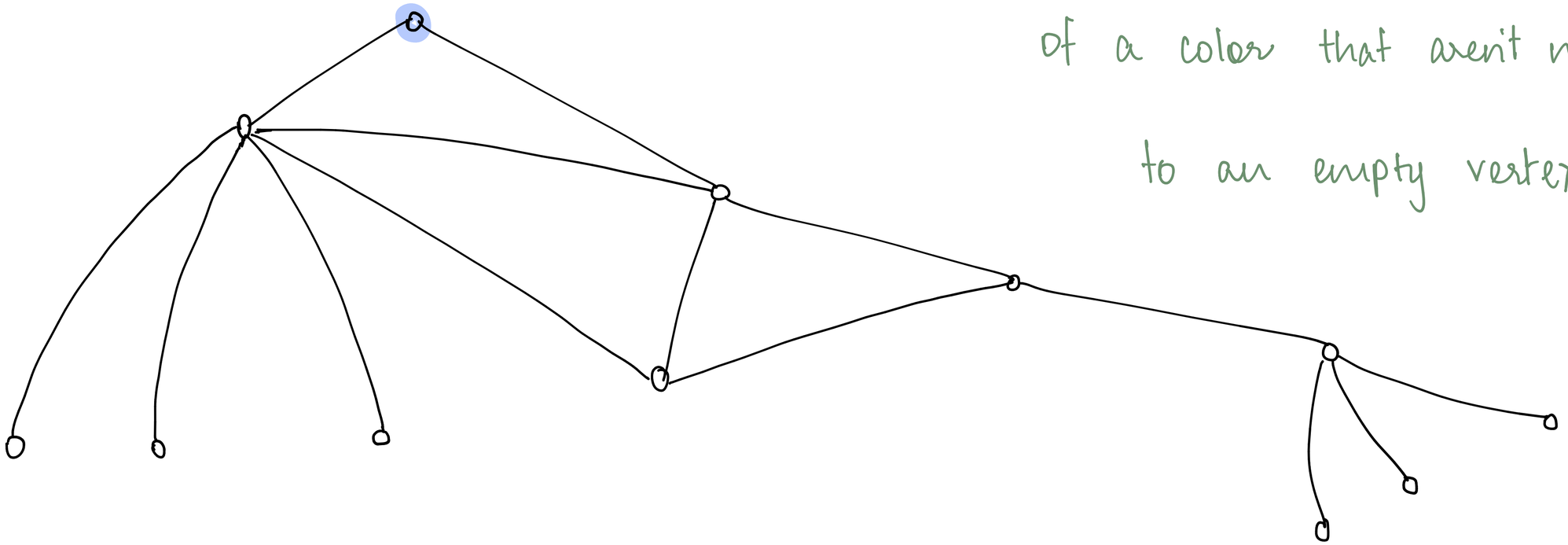
No Go

★ You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.



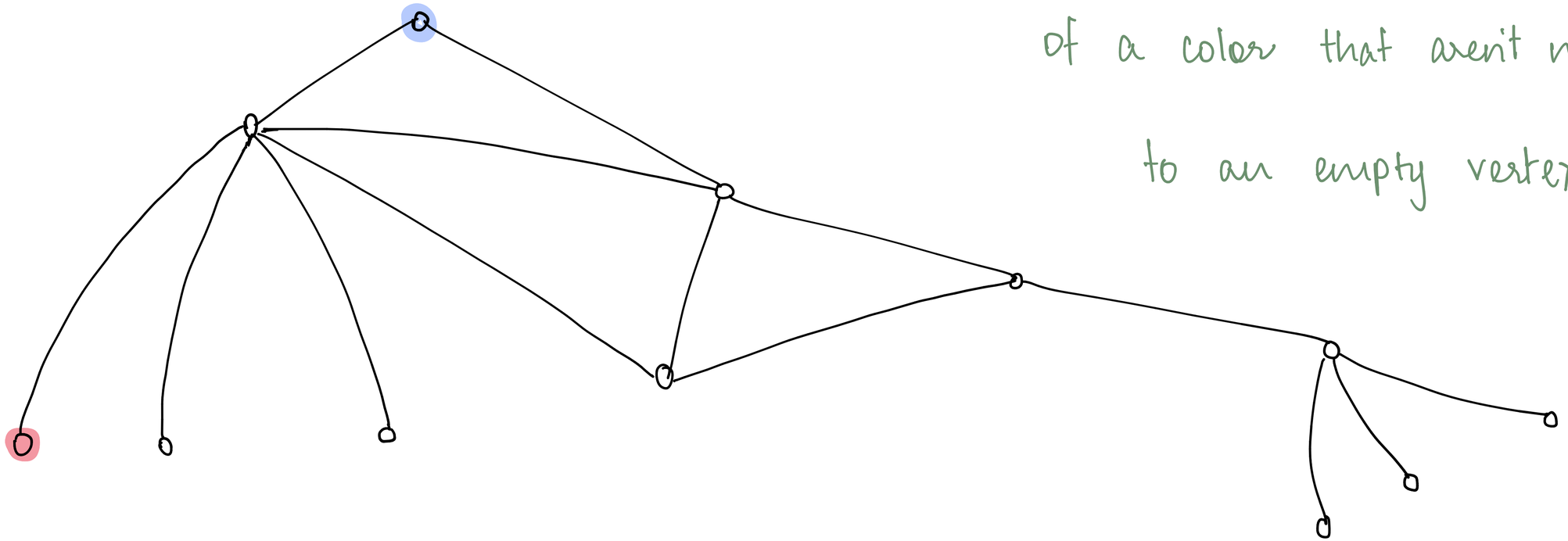
No Go

★ You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.



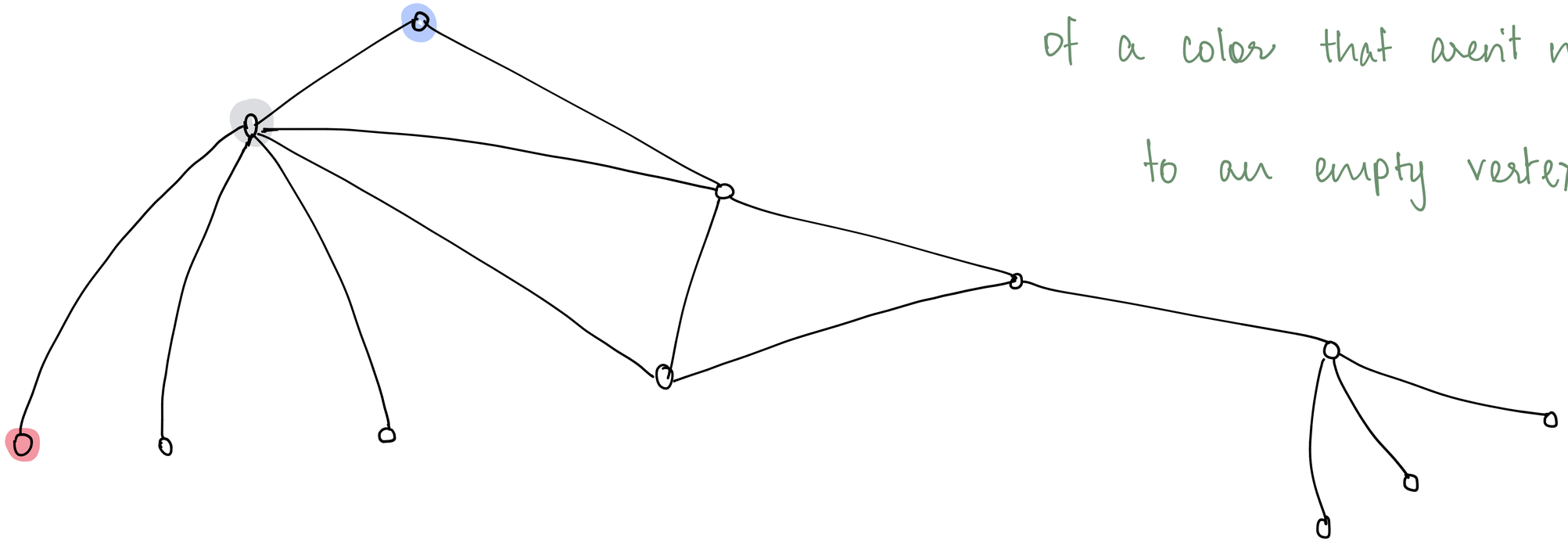
No Go

★ You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.



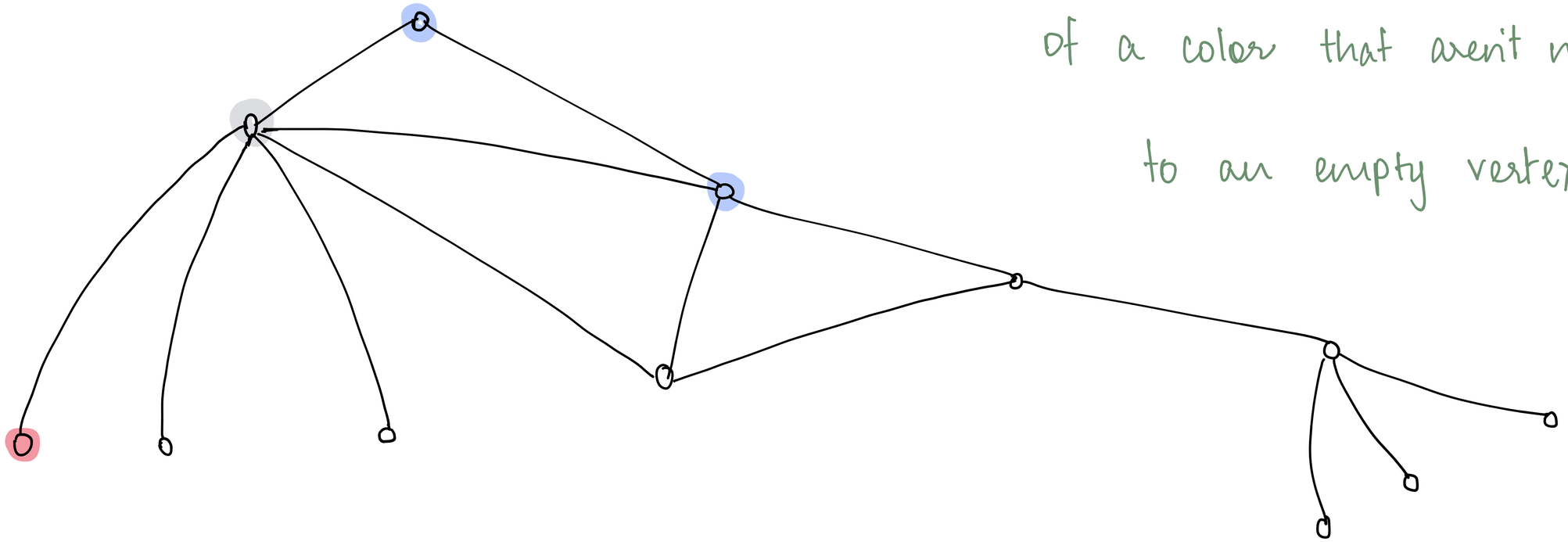
No Go

★ You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.



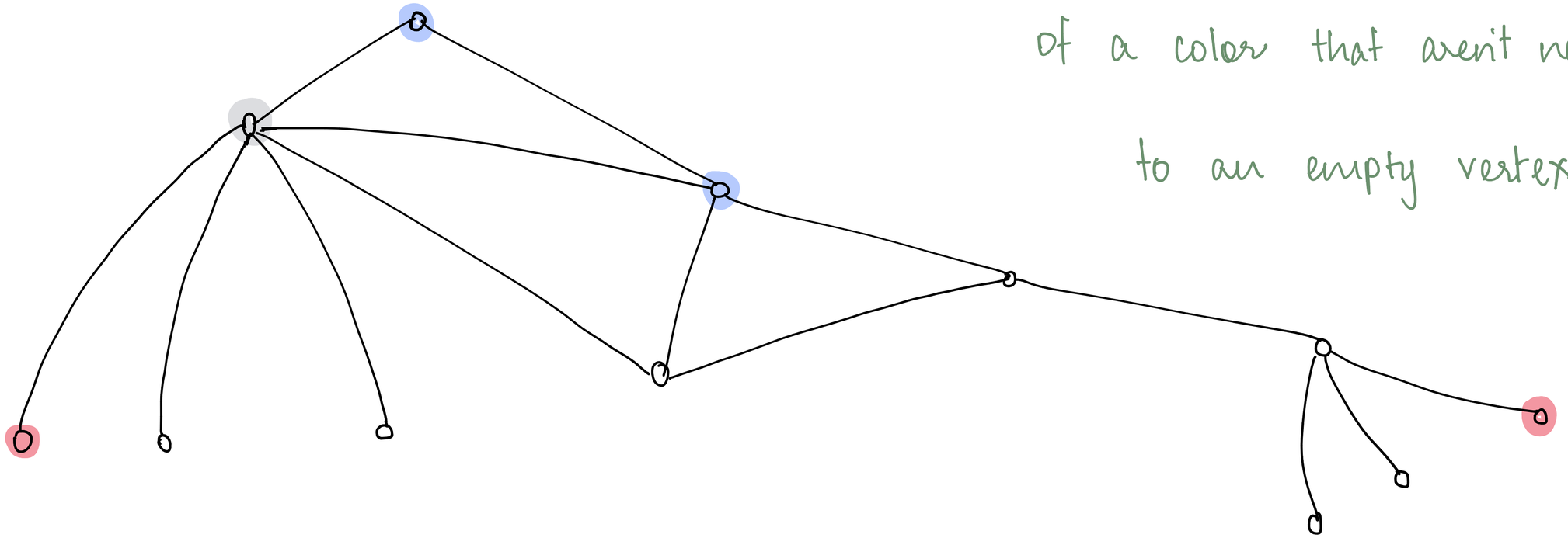
No Go

★ You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.



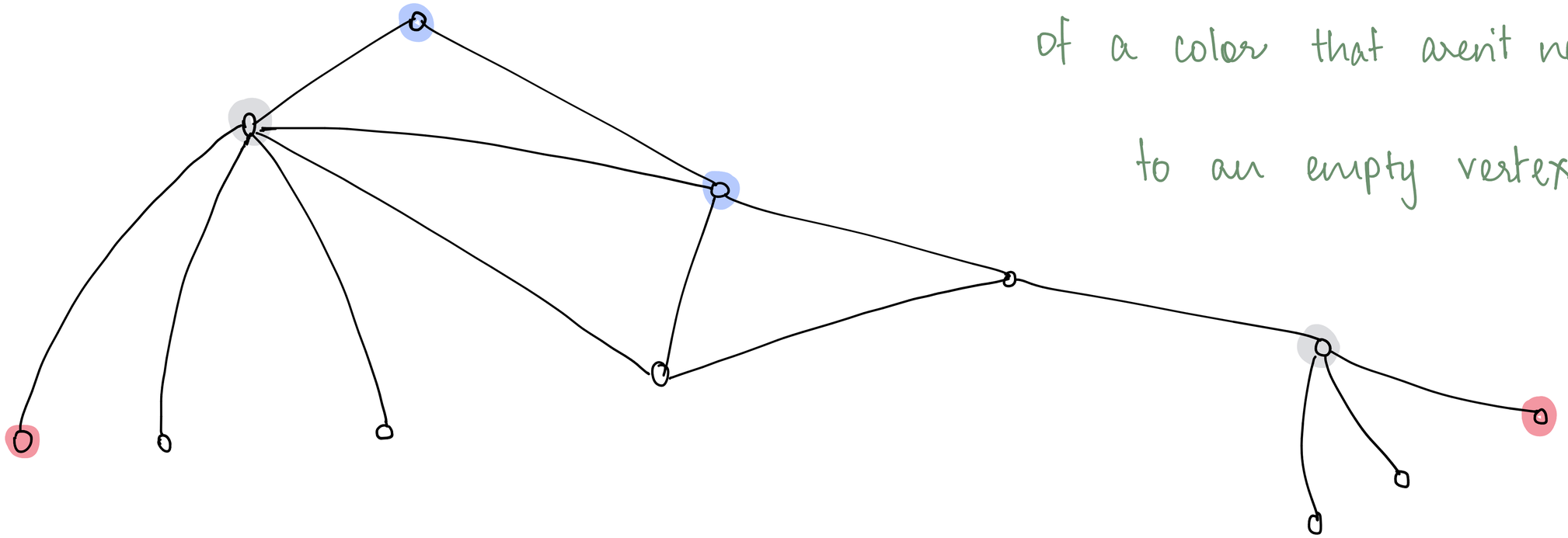
No Go

★ You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.



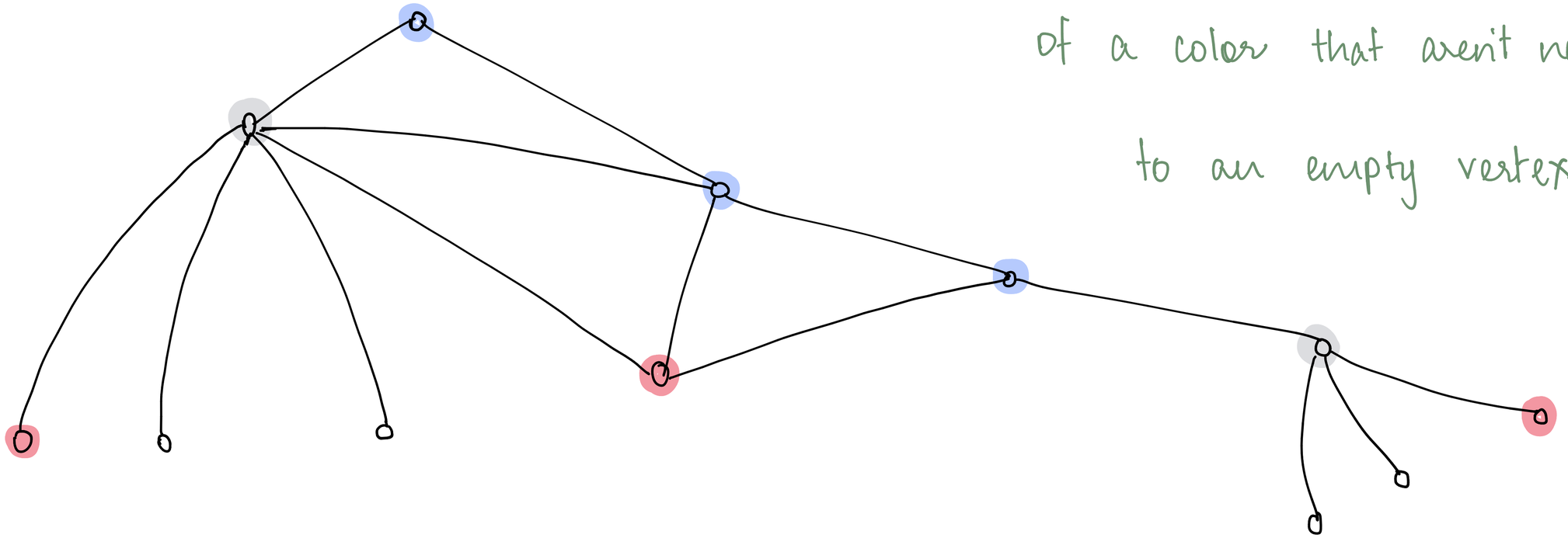
No Go

★ You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.



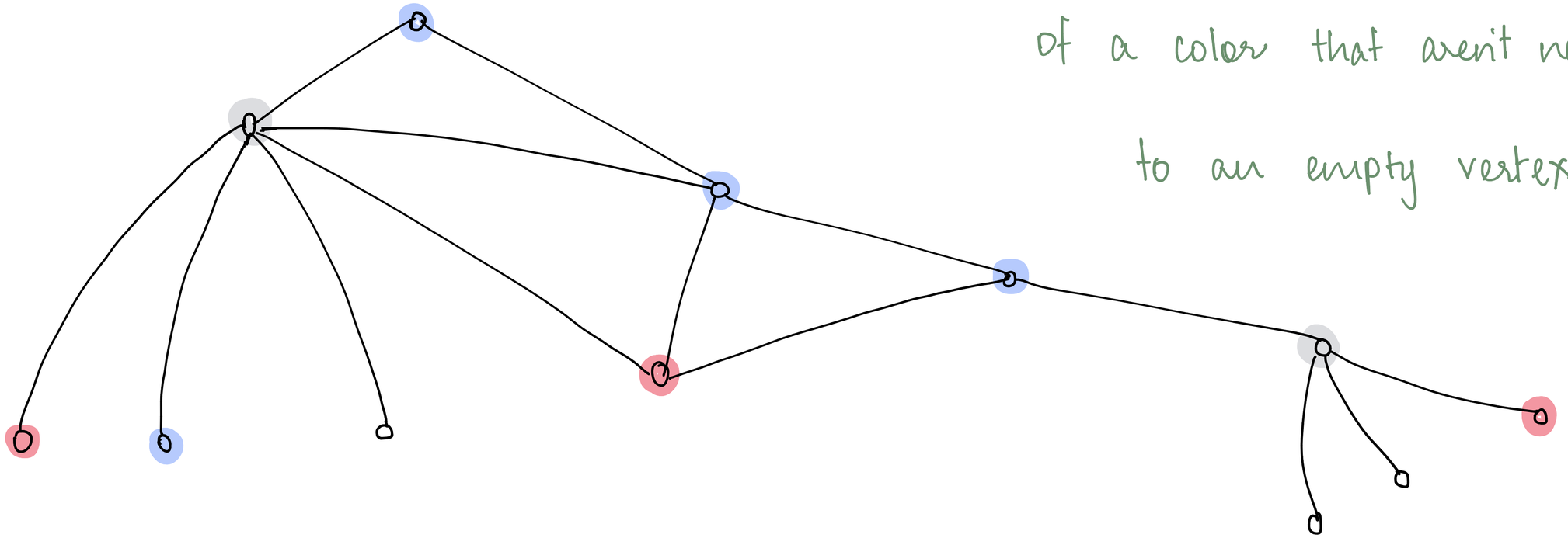
No Go

★ You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.



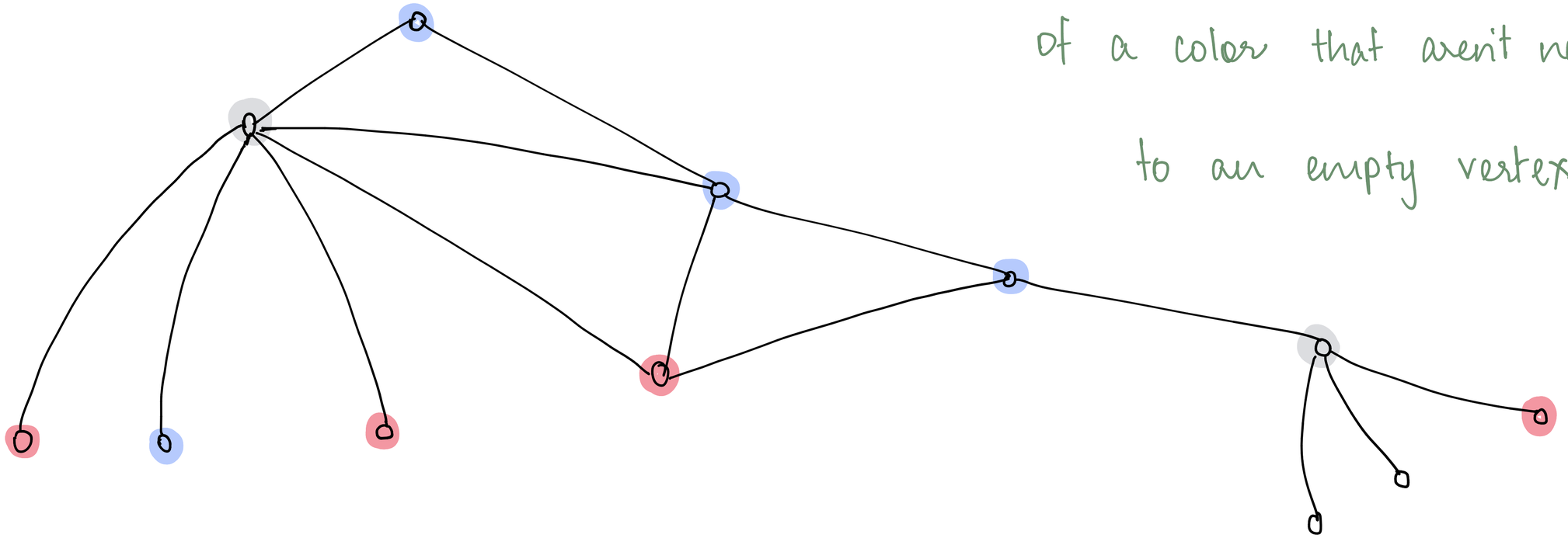
No Go

★ You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.



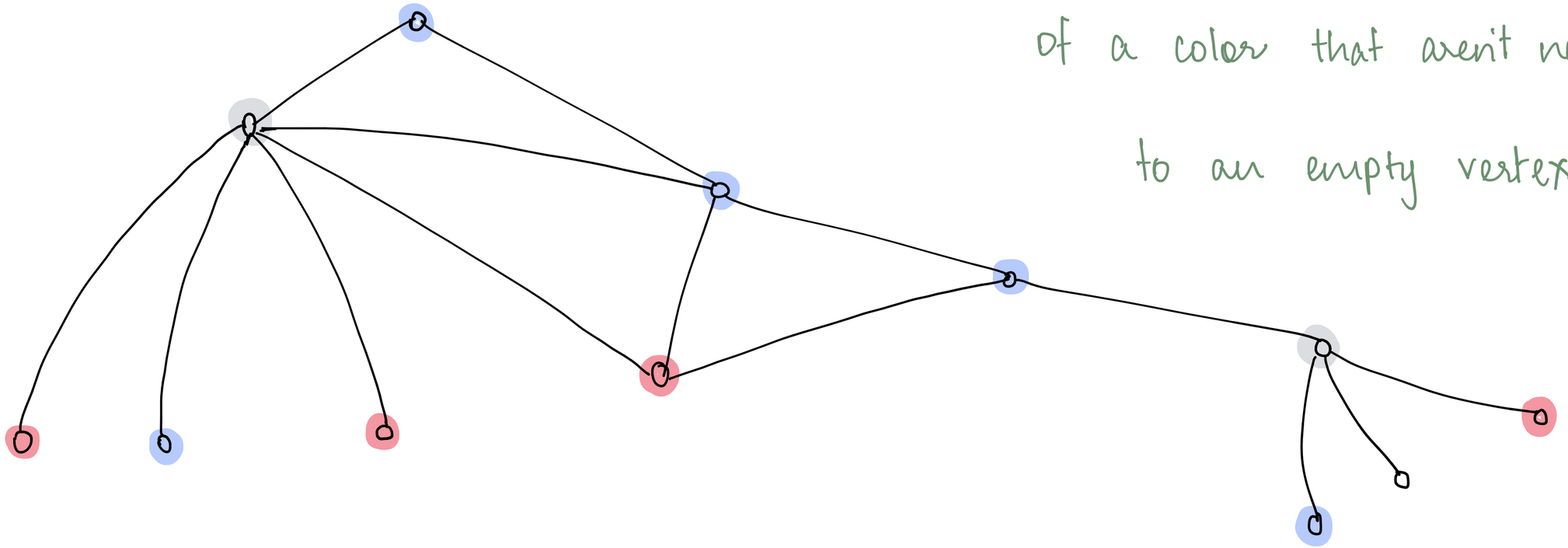
No Go

★ You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.



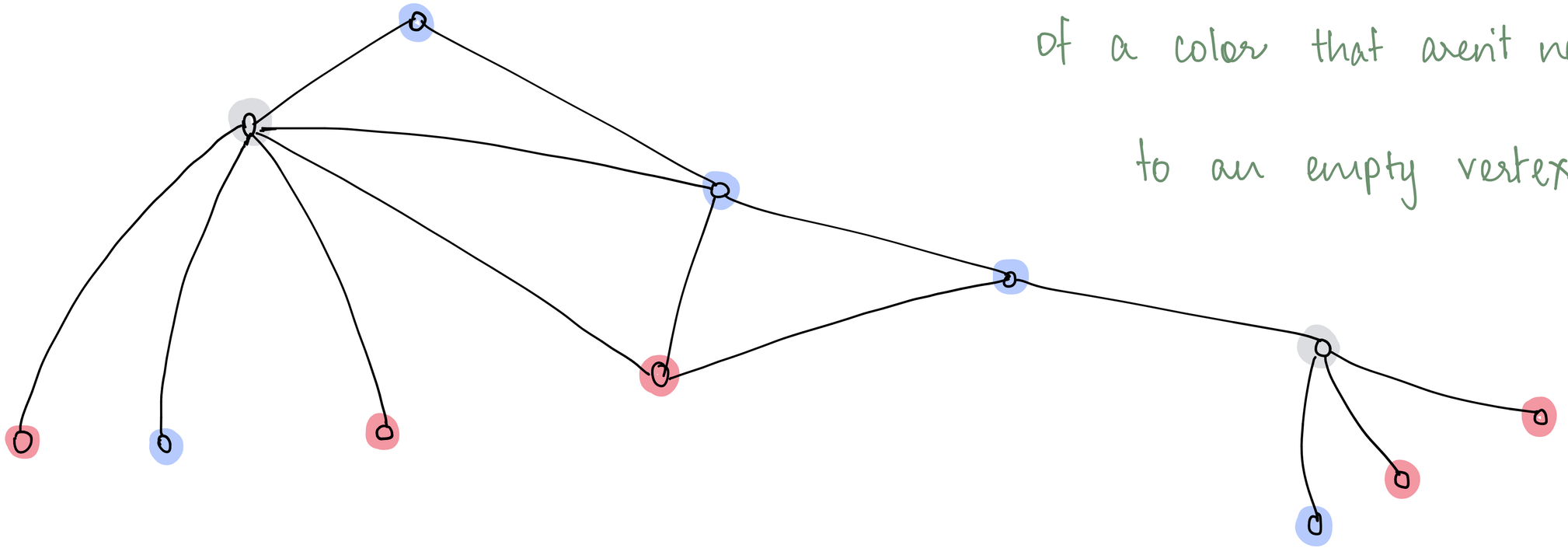
No Go

★ You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.



No Go

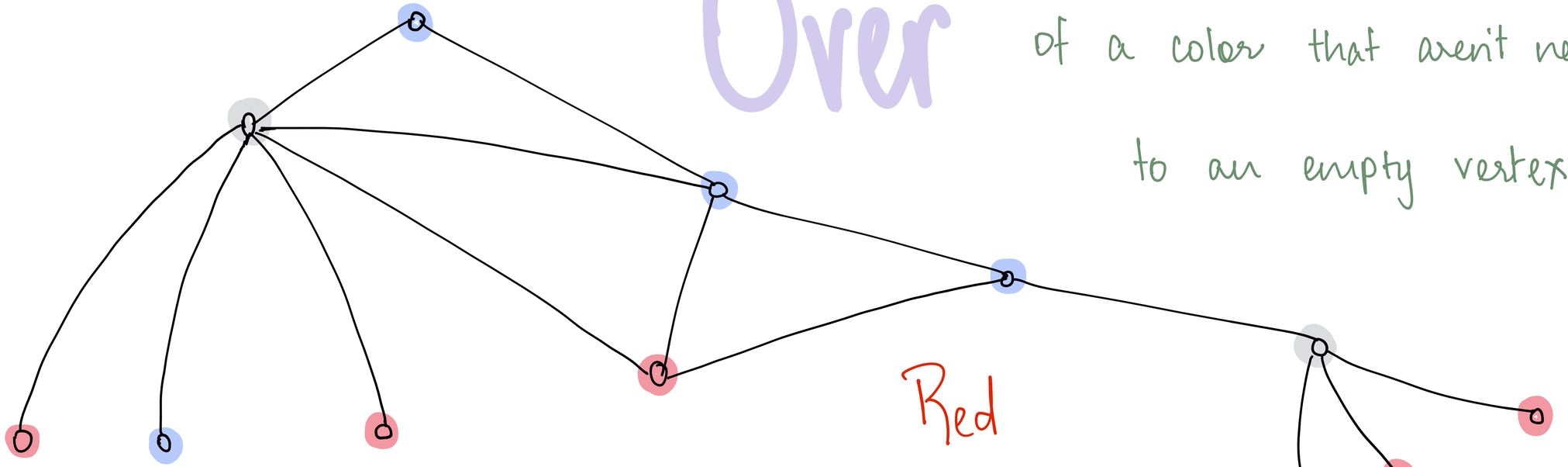
★ You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.



No Go

Game Over

★ You can't choose a vertex if it would create a board where there is a cc of vertices of a color that aren't next to an empty vertex.



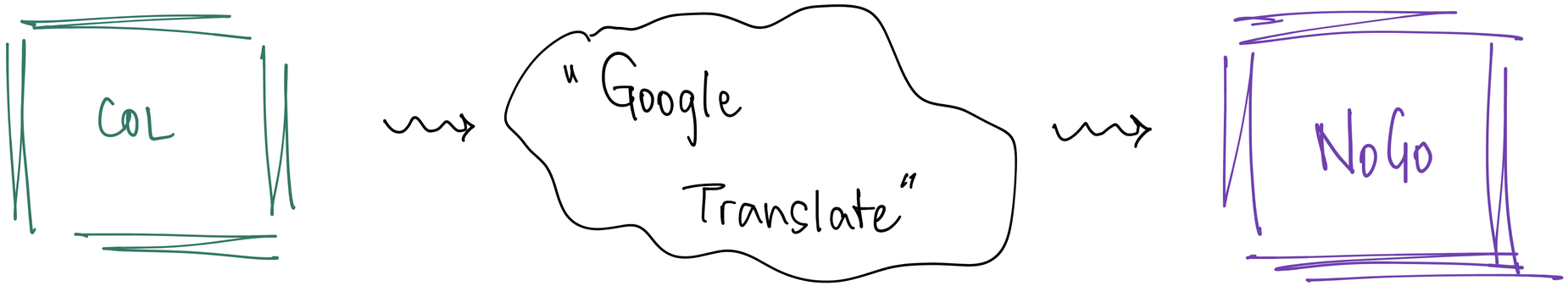
Red Wins

Your friend has challenged you to a game of GoL.

Your friend has challenged you to a game of GoL.

You can ask for help from a very smart & trusted AI

who is really good at playing NoGo.

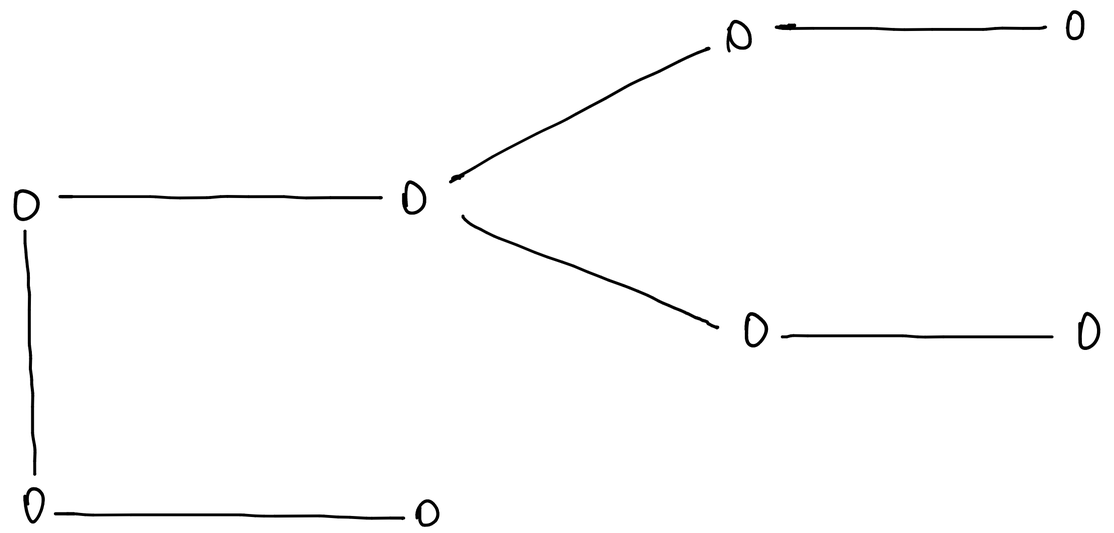


Your friend has challenged you to a game of CoL.

You can ask for help from a very smart & trusted AI who is really good at playing NoGo.

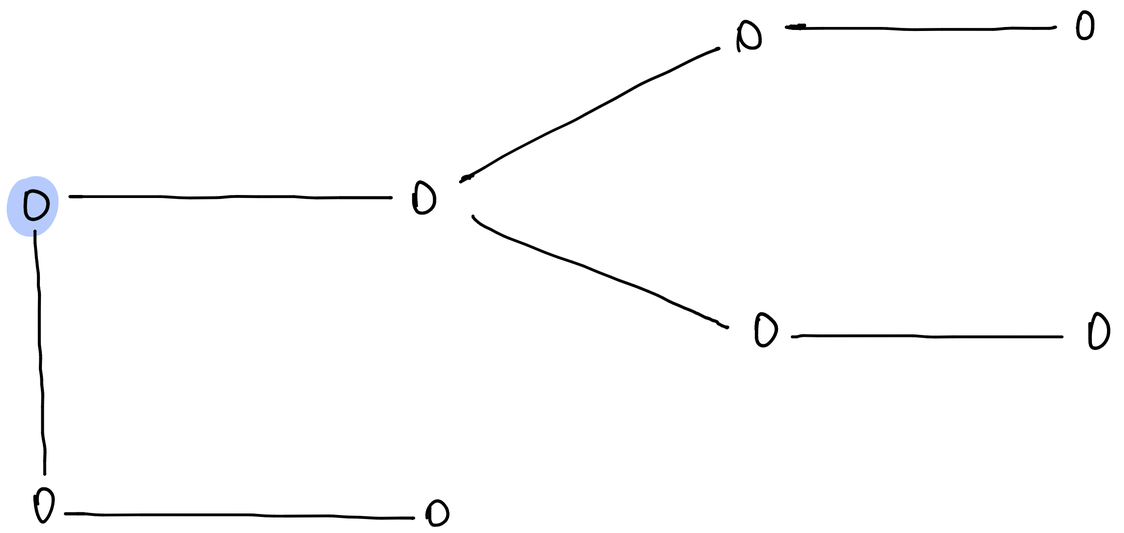
CoL $\leq$ NoGo

CoL: Cannot play at vertices adjacent to your own color.



Col $\leq$ NoGo

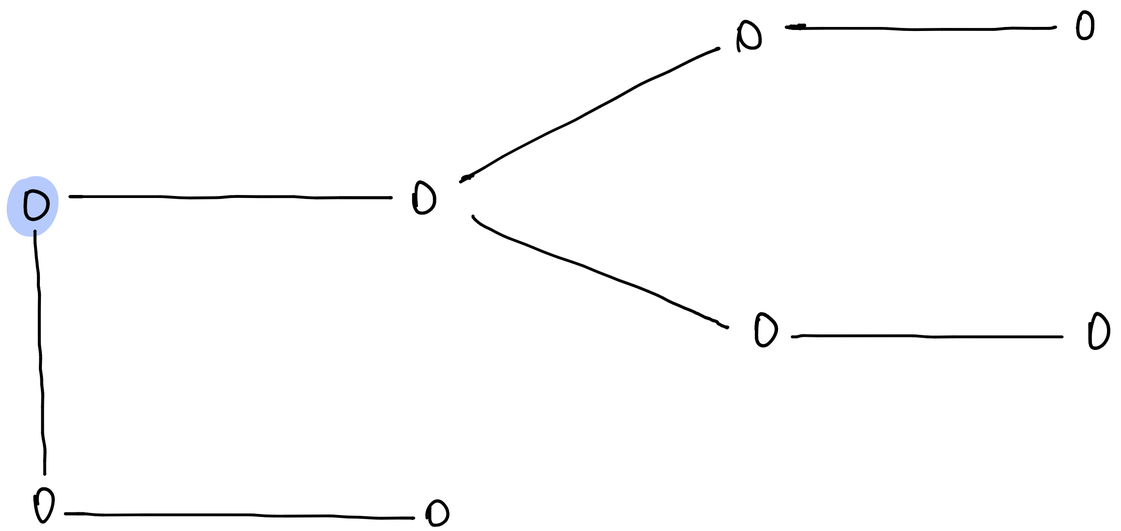
Col: Cannot play at vertices adjacent to your own color.



COL $\leq$ NOGO

COL: Cannot play at vertices adjacent to your own color.

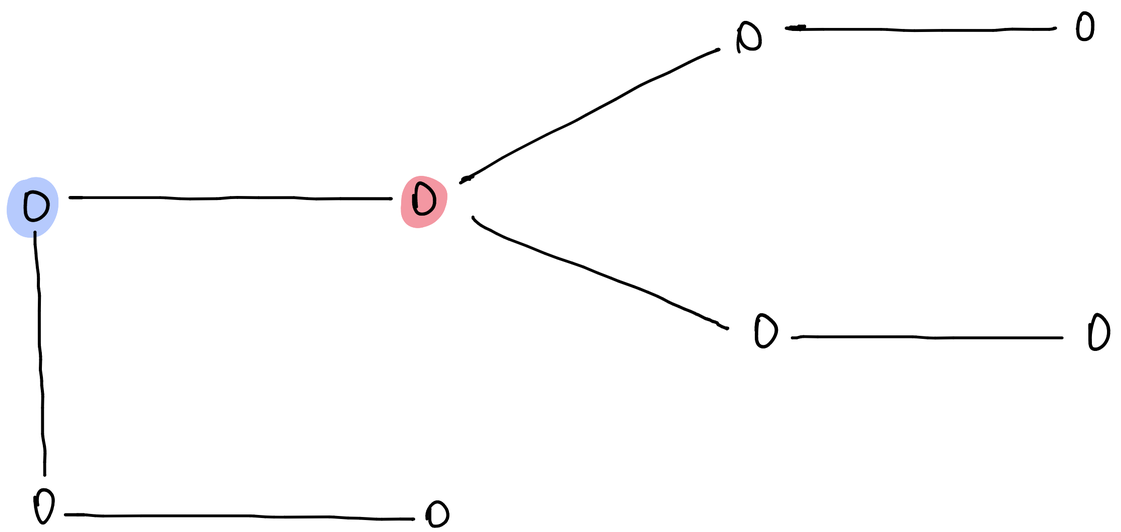
Idea: Just pretend this is a NOGO board & ping AI friend.



CoL $\leq$ NoGo

CoL: Cannot play at vertices adjacent to your own color.

Idea: Just pretend this is a NoGo board & ping AI friend.

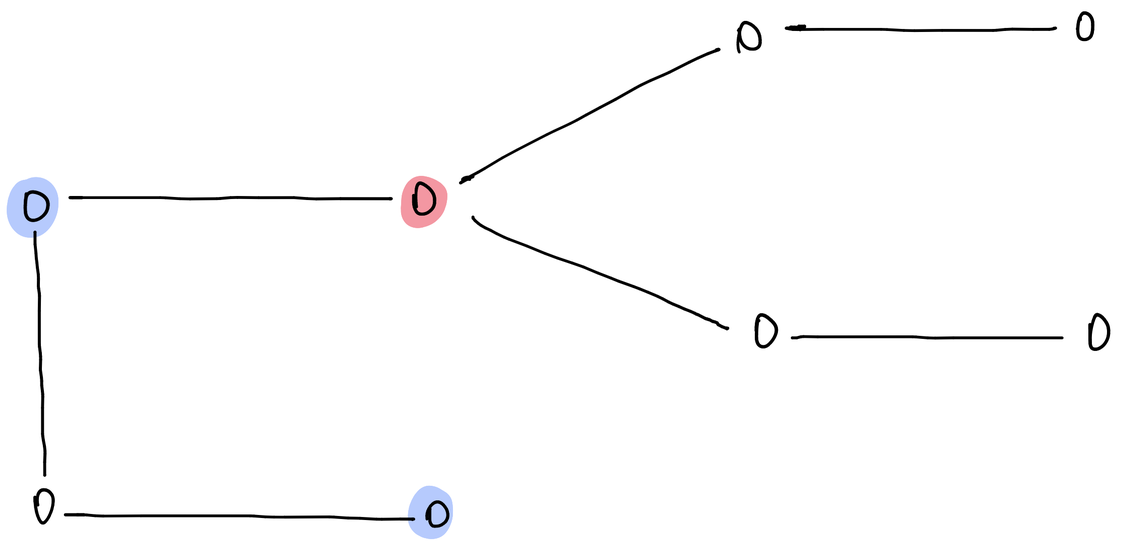


Issue: AI friend is oblivious to the rules of CoL & may violate them

CoL $\leq$ NoGo

CoL: Cannot play at vertices adjacent to your own color.

Idea: Just pretend this is a NoGo board & ping AI friend.

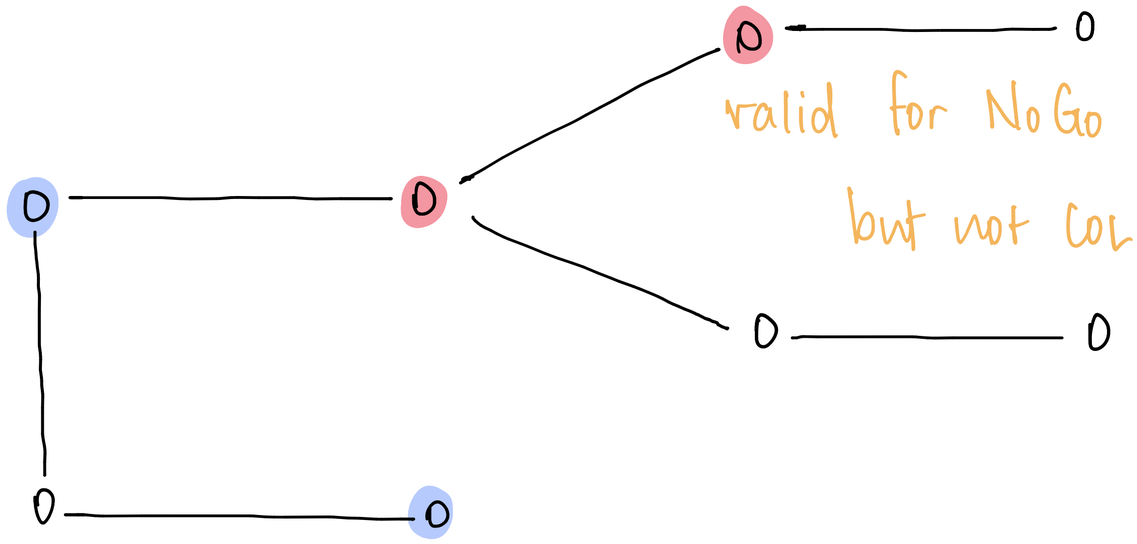


Issue: AI friend is oblivious to the rules of CoL & may violate them

CoL ≤ NoGo

CoL: Cannot play at vertices adjacent to your own color.

Idea: Just pretend this is a NoGo board & ping AI friend.



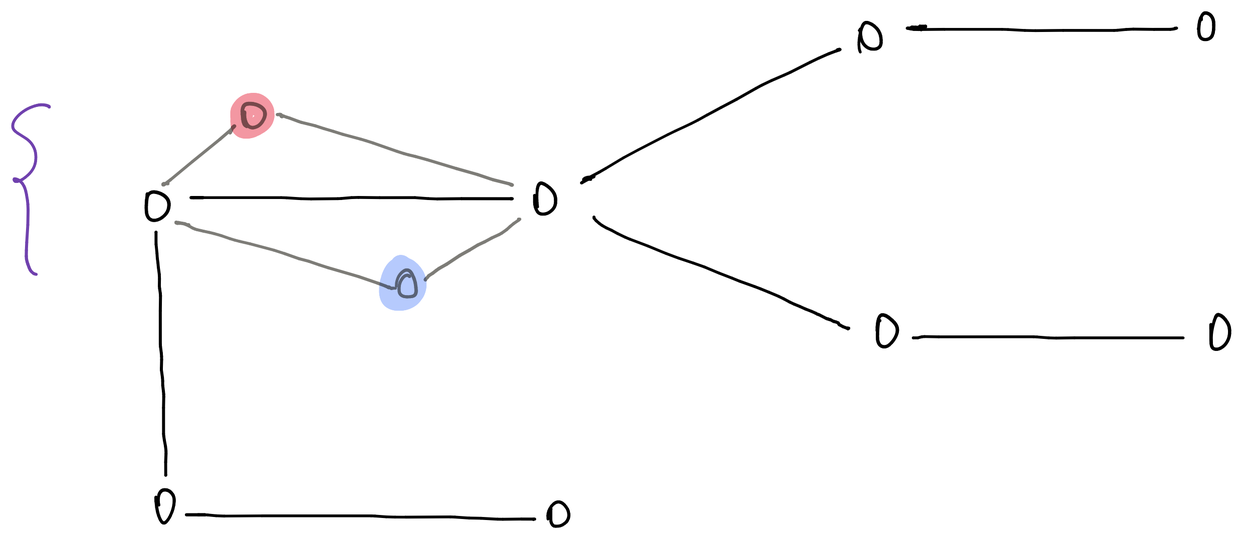
valid for NoGo but not CoL.

Issue: AI friend is oblivious to the rules of CoL & may violate them

Col $\leq$ NoGo

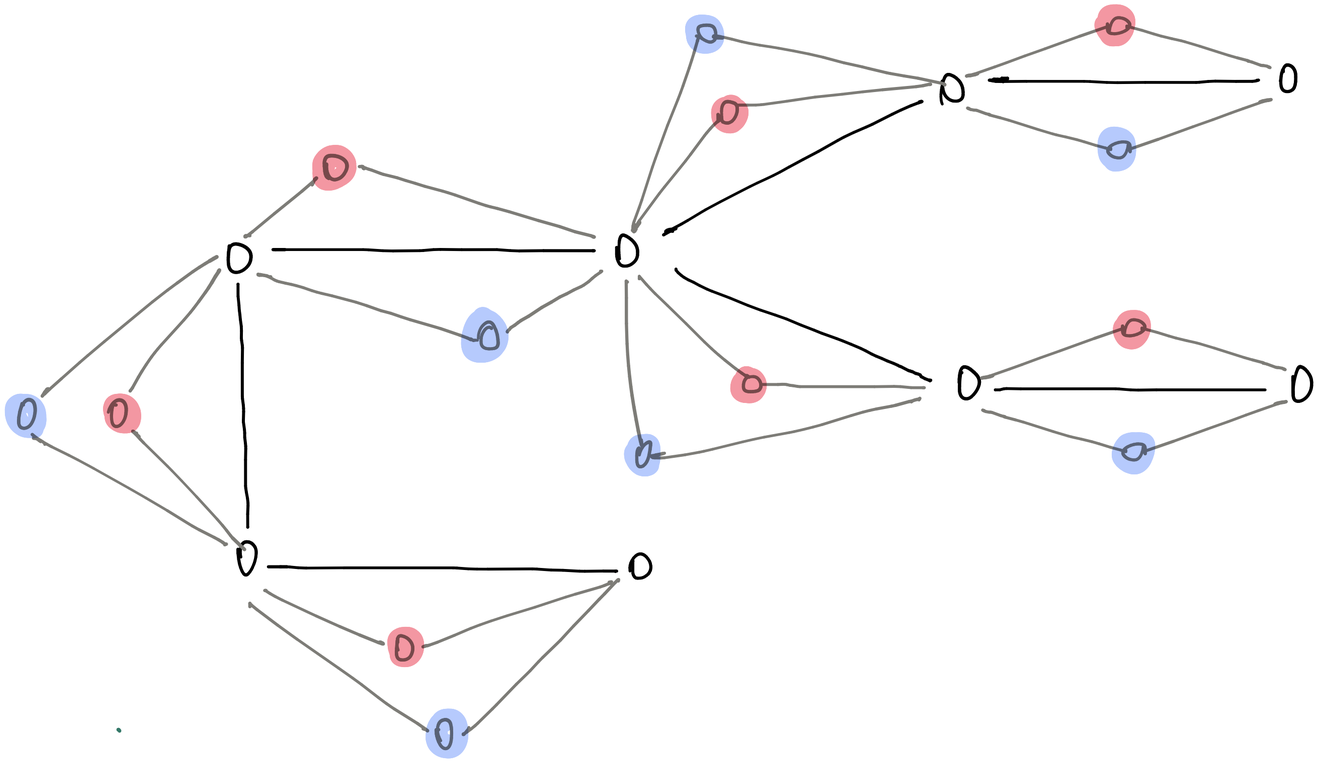
Col: Cannot play at vertices adjacent to your own color.

apply this "hack" to EVERY edge.



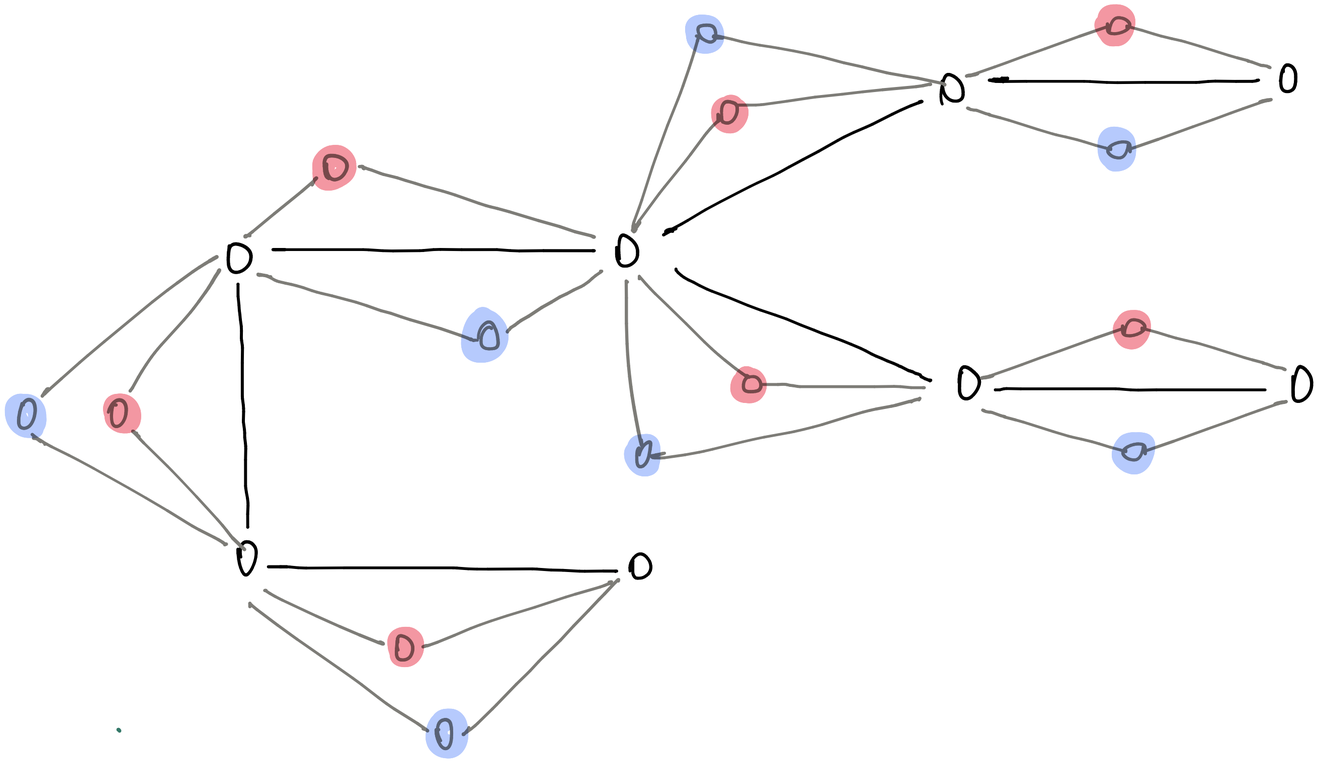
Col $\leq$ NoGo

Col: Cannot play at vertices adjacent to your own color.



CoL $\leq$ NoGo

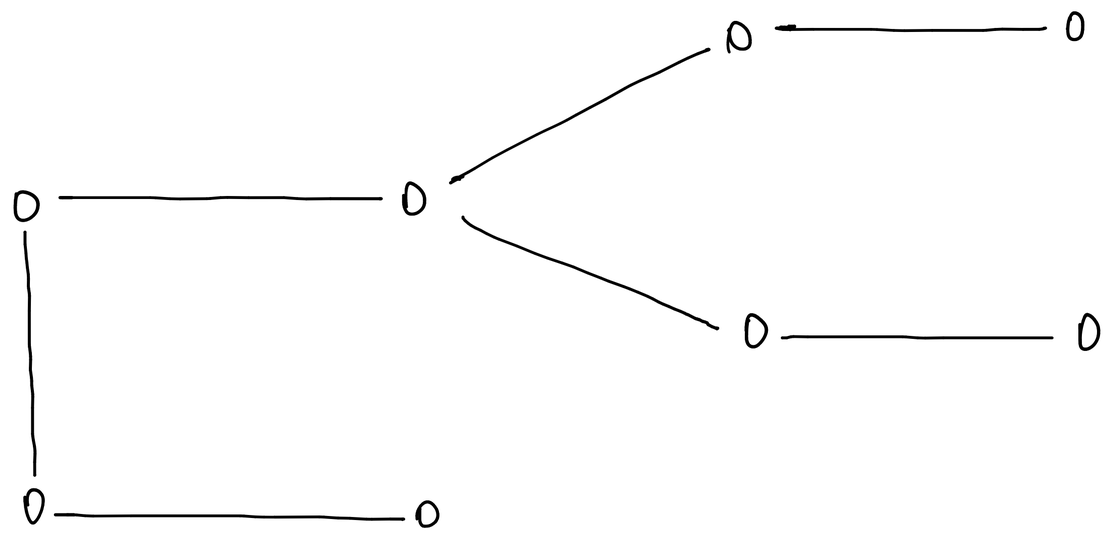
CoL: Cannot play at vertices adjacent to your own color.



NoGo on this board simulates

$\text{Col} \leq \text{NoGo}$

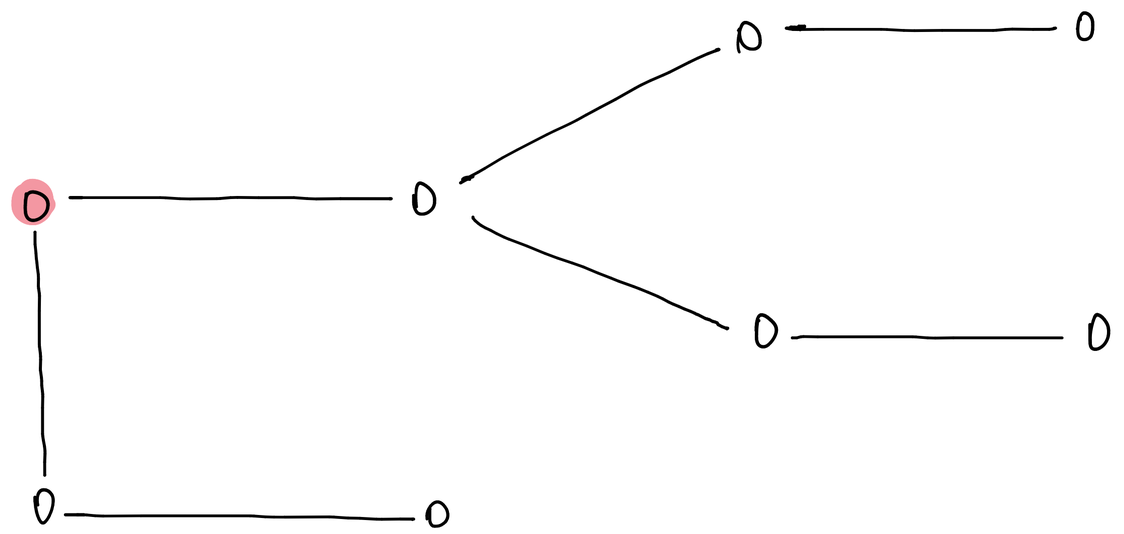
Col: Cannot play at vertices adjacent to your own color.



Col on the original

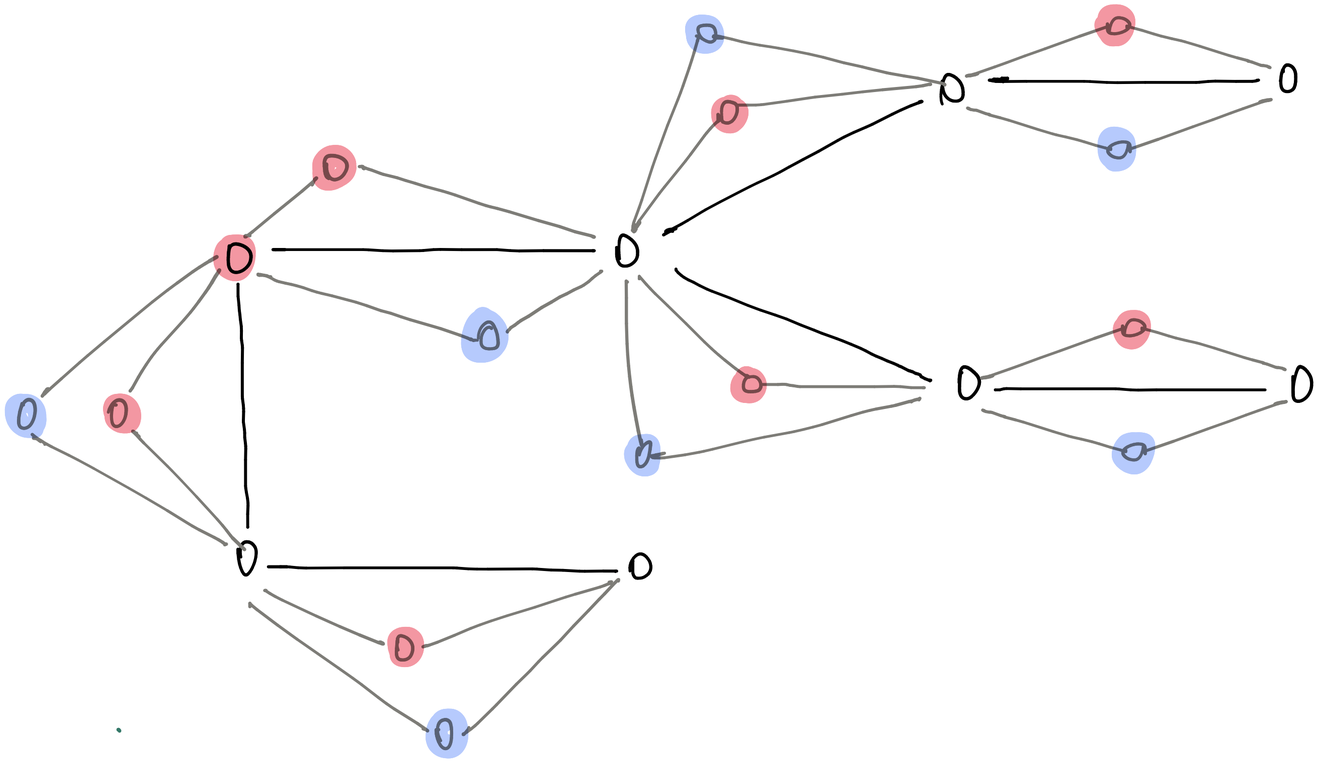
Col $\leq$ NoGo

Col: Cannot play at vertices adjacent to your own color.



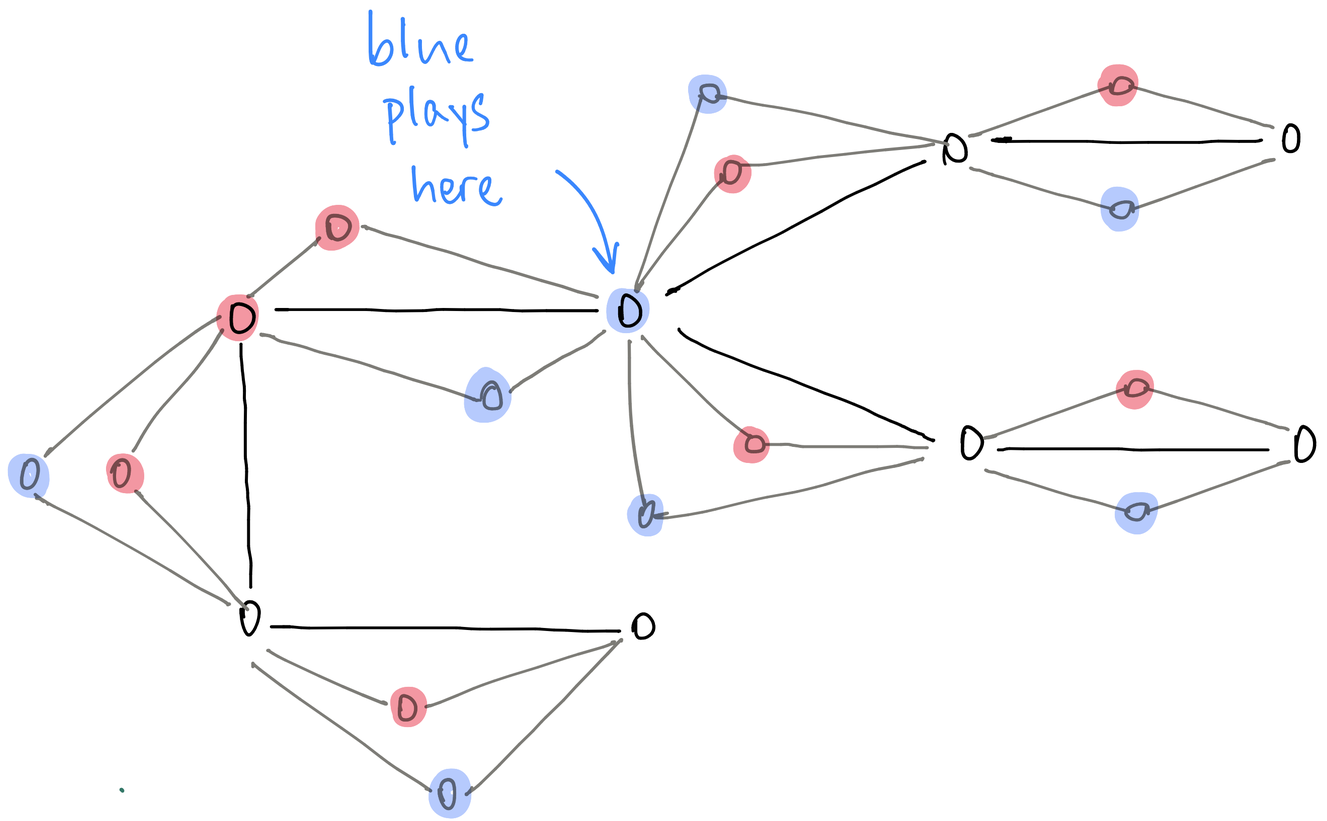
Col $\leq$ NoGo

Col: Cannot play at vertices adjacent to your own color.



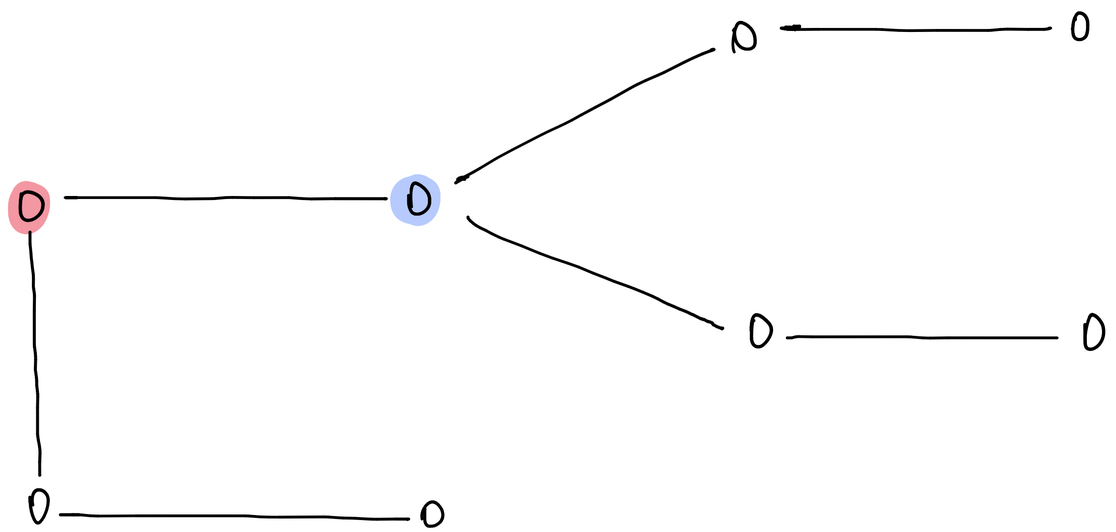
Col $\leq$ NoGo

Col: Cannot play at vertices adjacent to your own color.



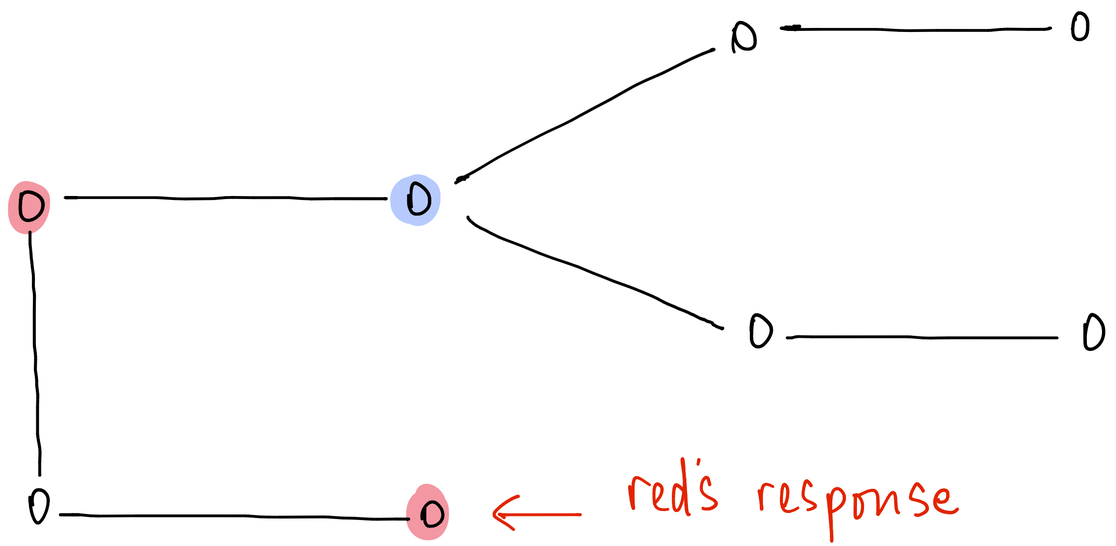
CoL $\leq$ NoGo

CoL: Cannot play at vertices adjacent to your own color.



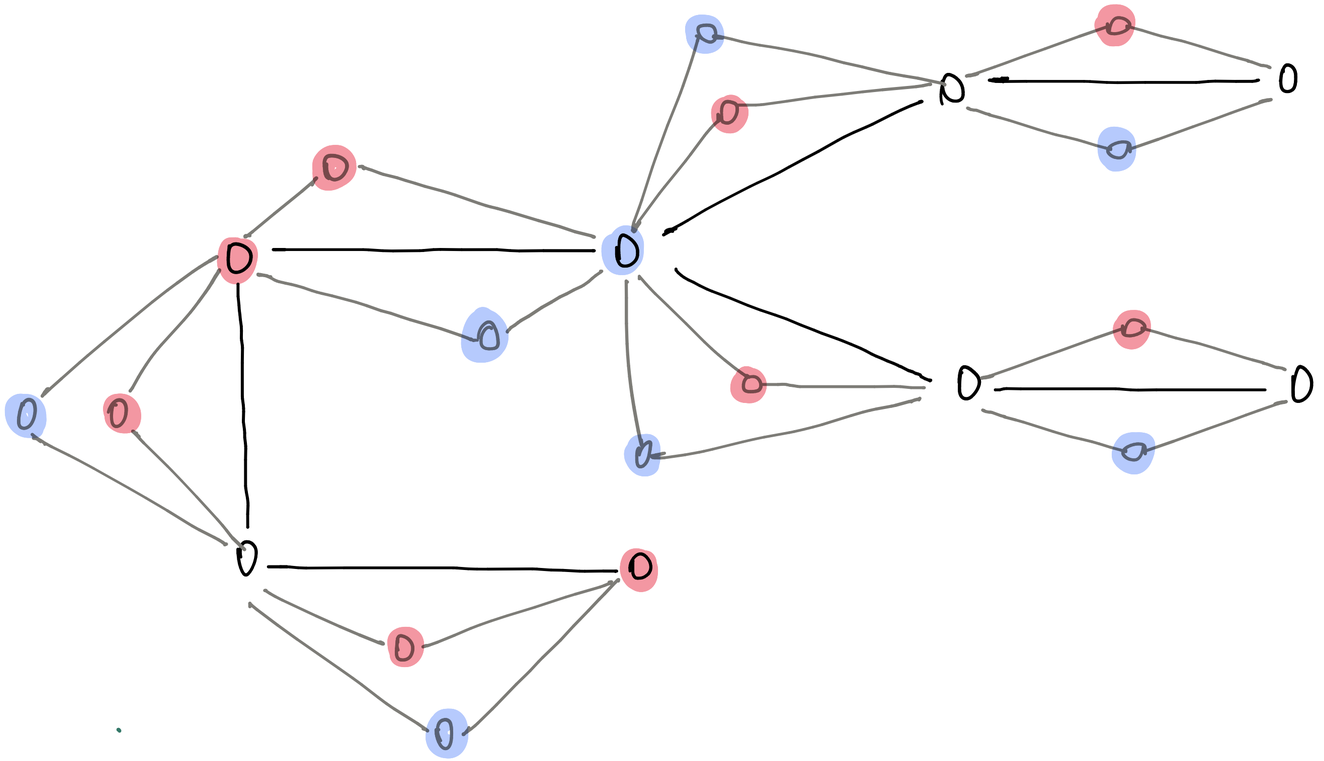
CoL $\leq$ NoGo

CoL: Cannot play at vertices adjacent to your own color.



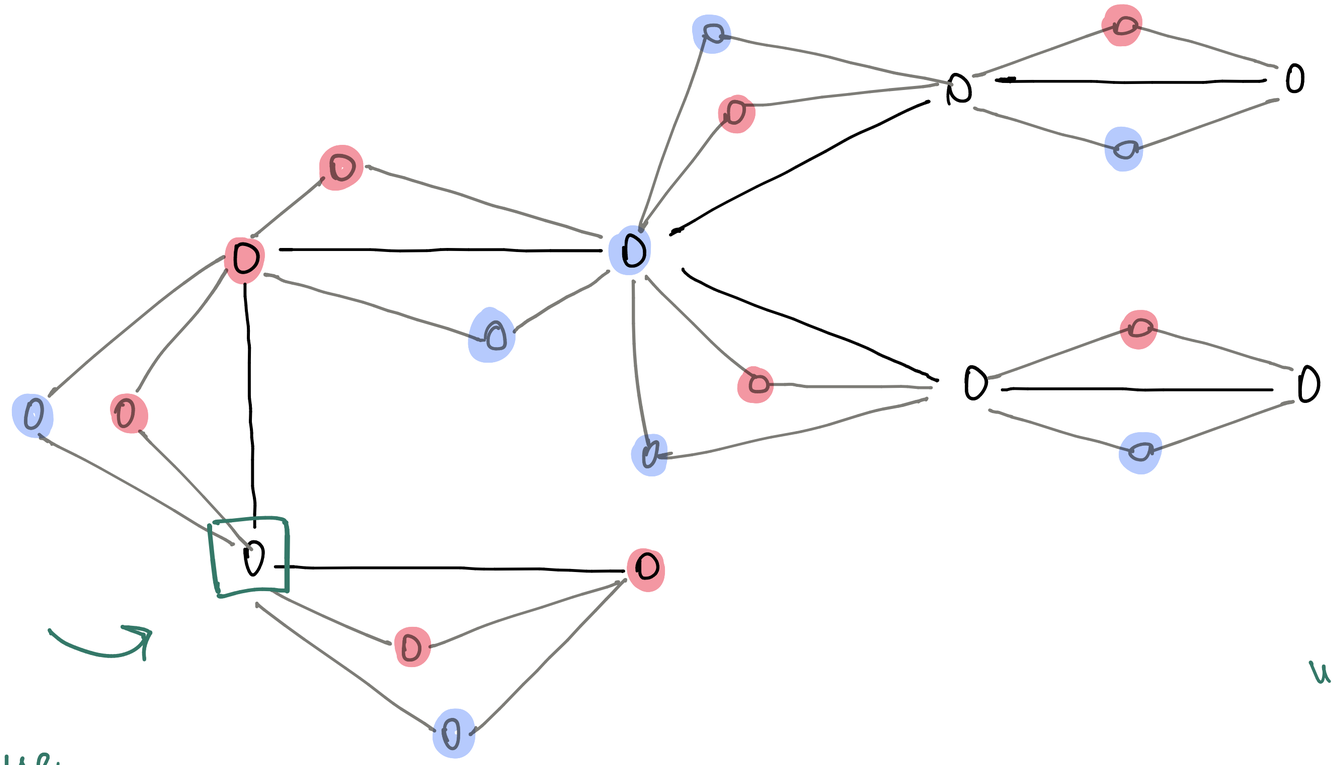
Col $\leq$ NoGo

Col: Cannot play at vertices adjacent to your own color.



Col ≤ NoGo

Col: Cannot play at vertices adjacent to your own color.



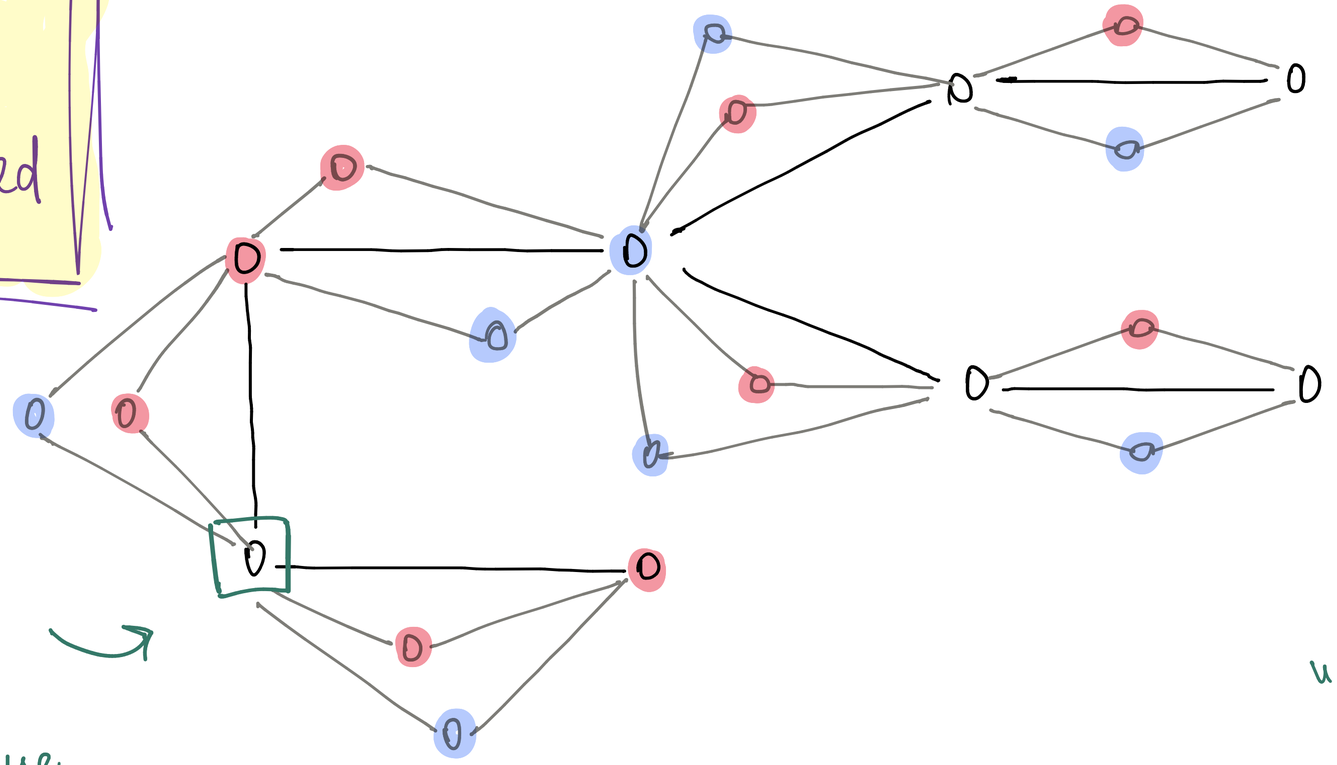
now blocked for blue

even though this move was perfectly valid in the "base" Col graph.

$\text{Col} \leq \text{NoGo}$

Col: Cannot play at vertices adjacent to your own color.

our NoGo board is over-constrained

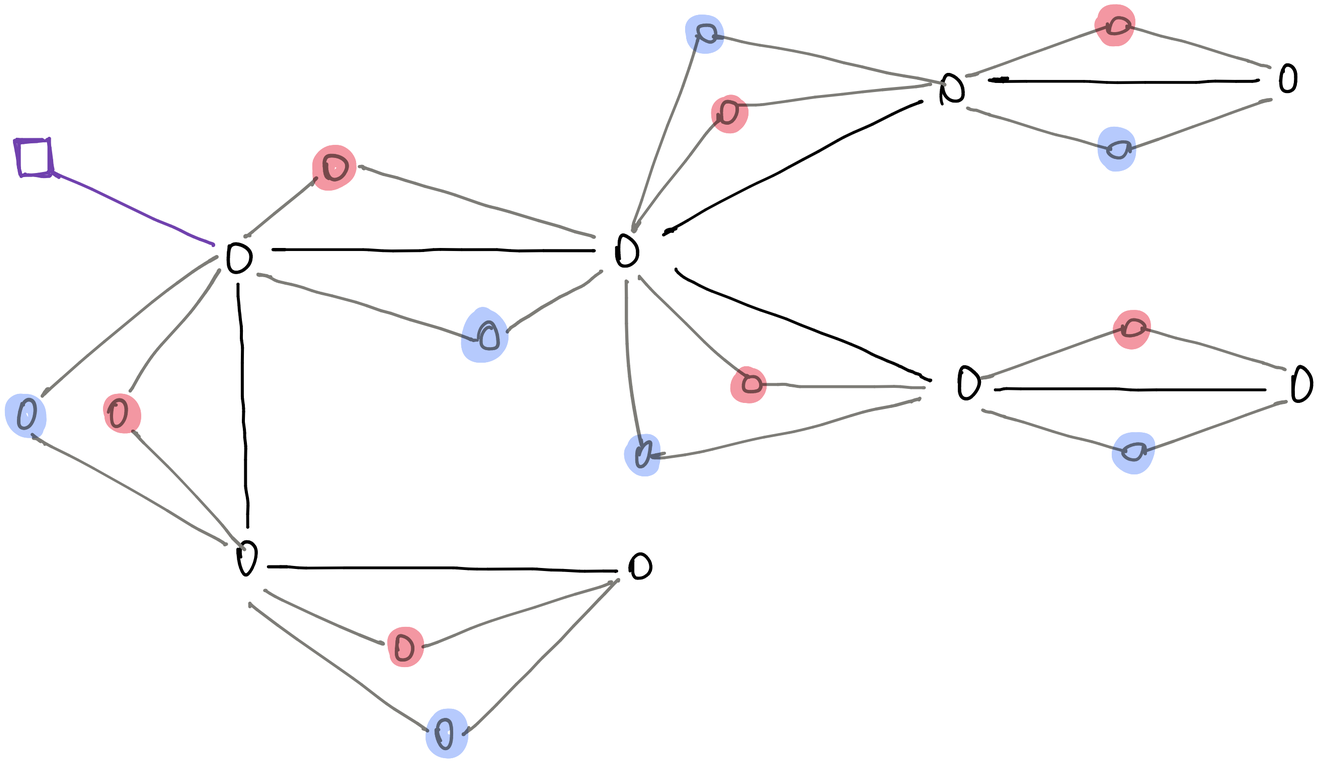


now blocked for blue

even though this move was perfectly valid in the "base" Col graph.

Col ≤ NoGo

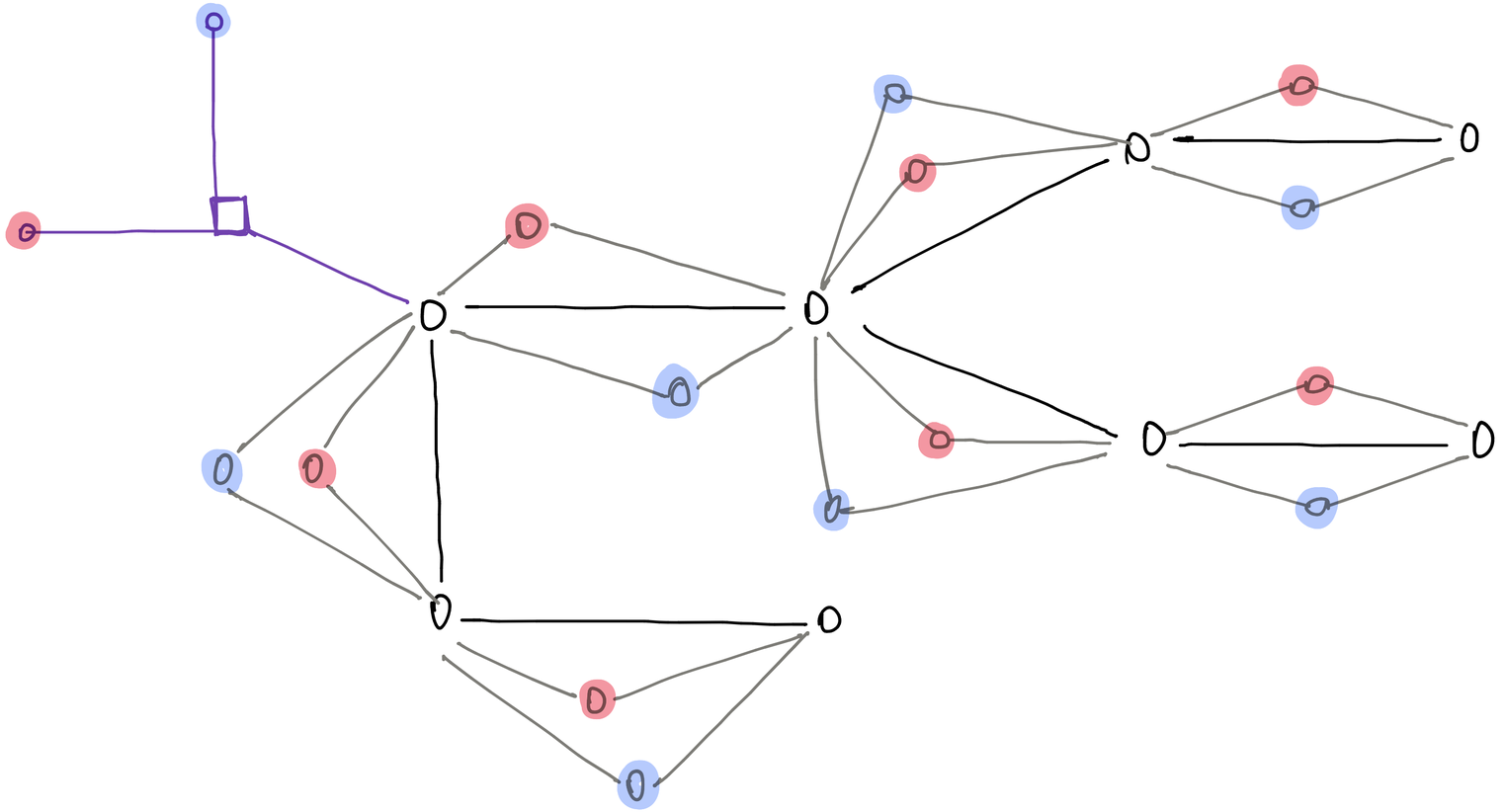
Col: Cannot play at vertices adjacent to your own color.



how can we "ensure" that □ remains blank always?

CoL $\leq$ NoGo

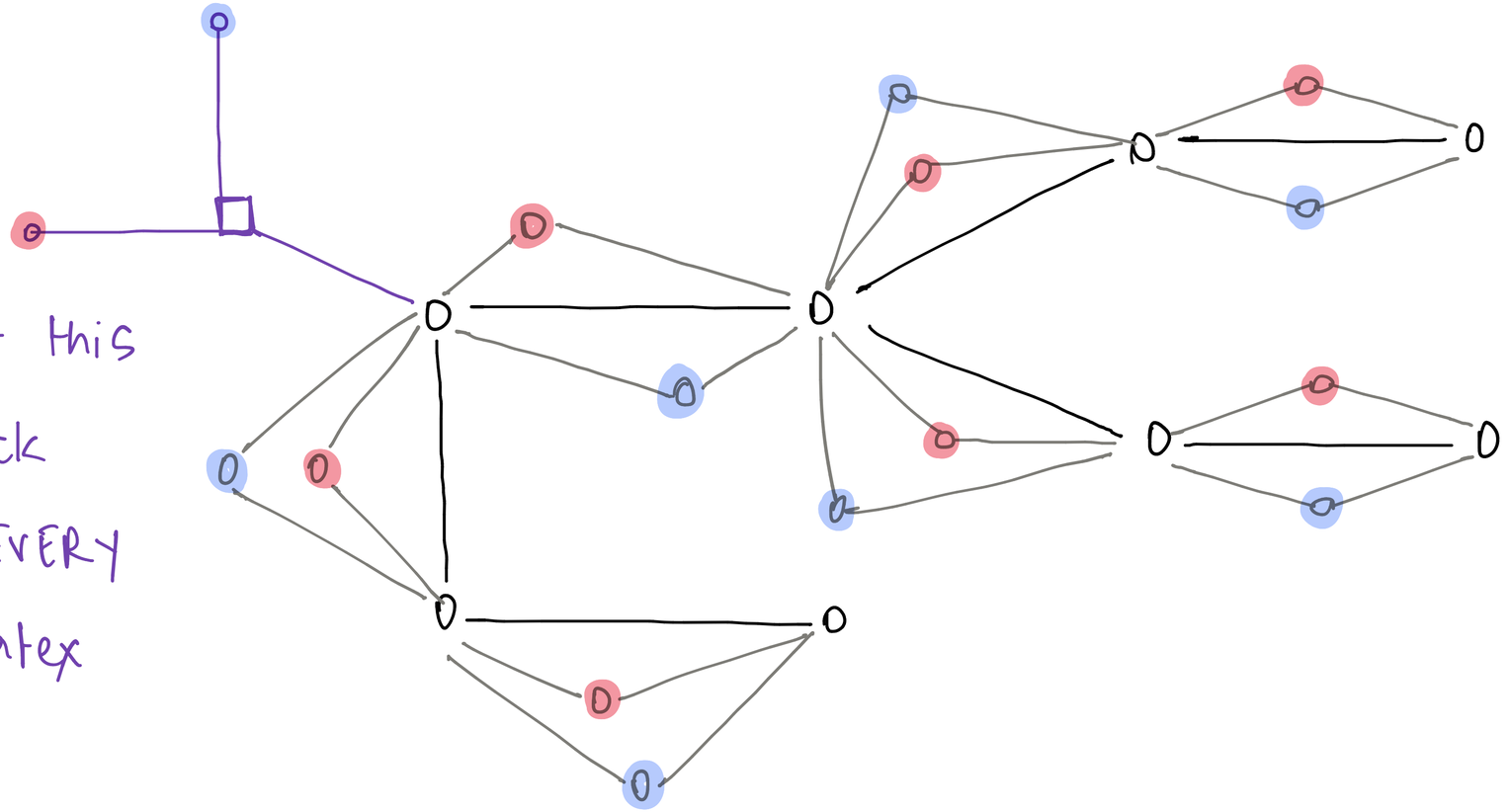
CoL: Cannot play at vertices adjacent to your own color.



how can we "ensure" that □ remains blank always?

Col ≤ NoGo

Col: Cannot play at vertices adjacent to your own color.



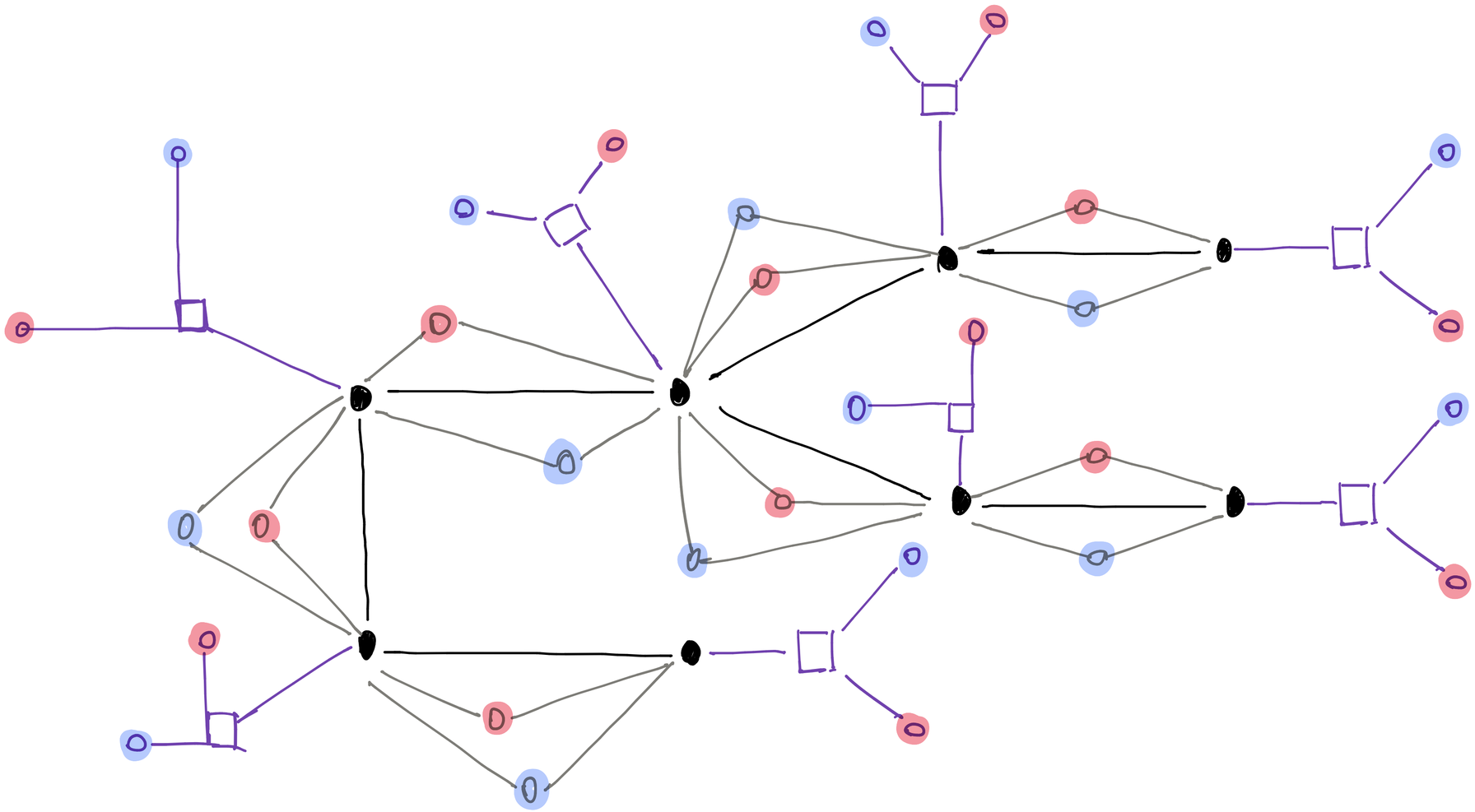
repeat this hack for EVERY vertex

how can we "ensure"

that □ remains

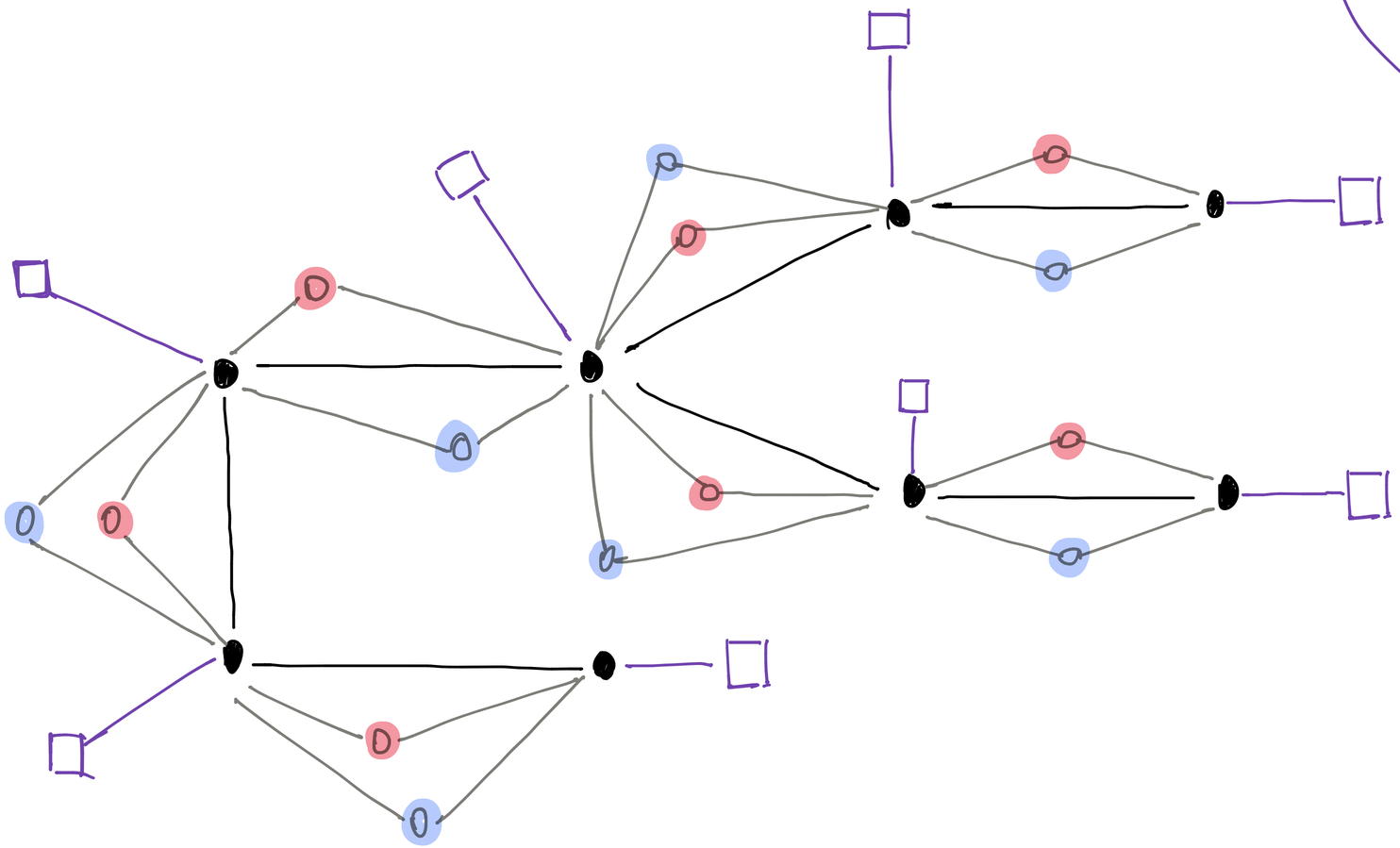
blank always?

$\mathrm{CoL} \leq \mathrm{NoGo}$



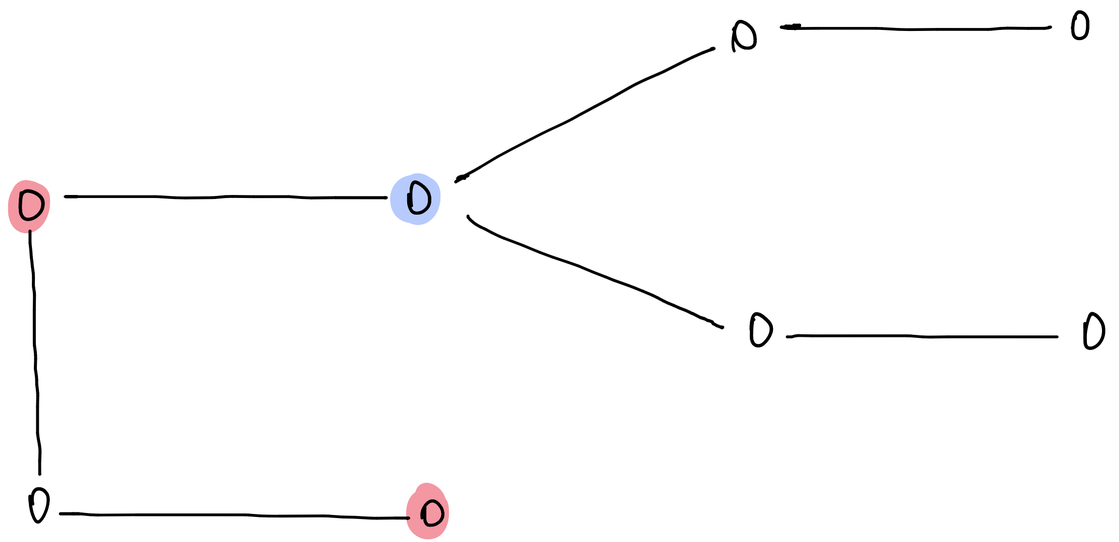
CoL ≤ NoGo

□ abbreviates

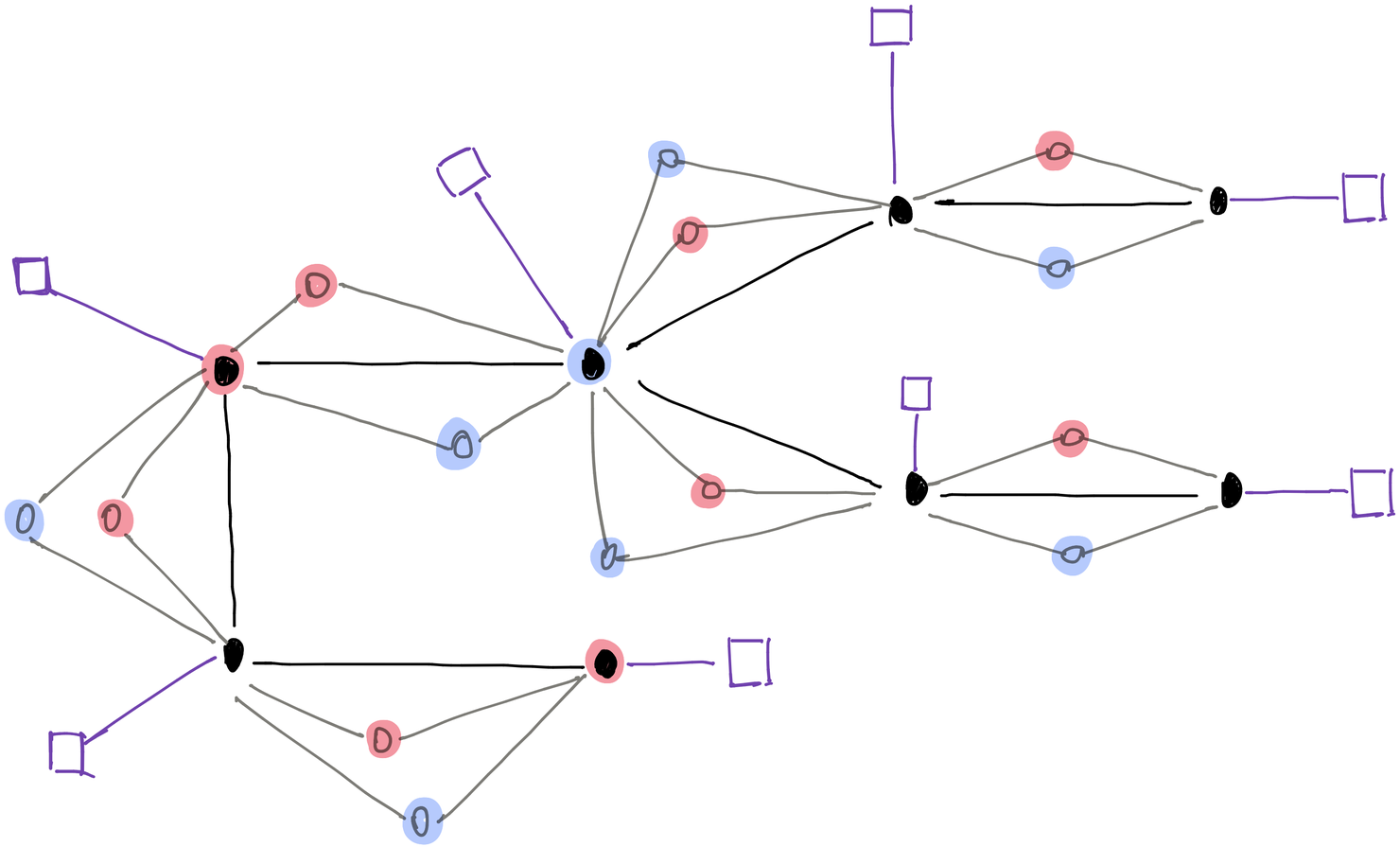


CoL $\leq$ NoGo

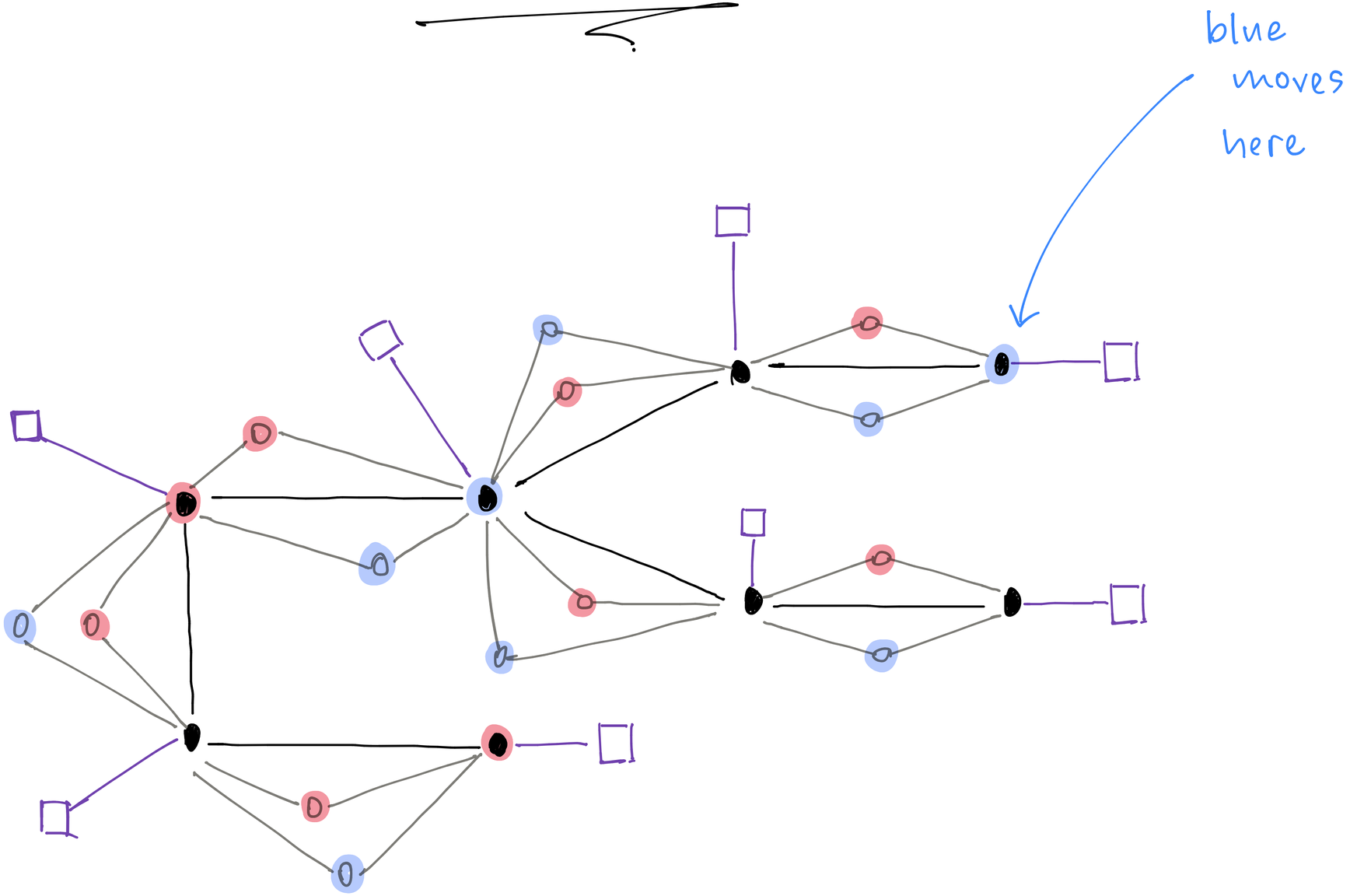
CoL: Cannot play at vertices adjacent to your own color.



$\mathrm{Col} \leq \mathrm{NoGo}$

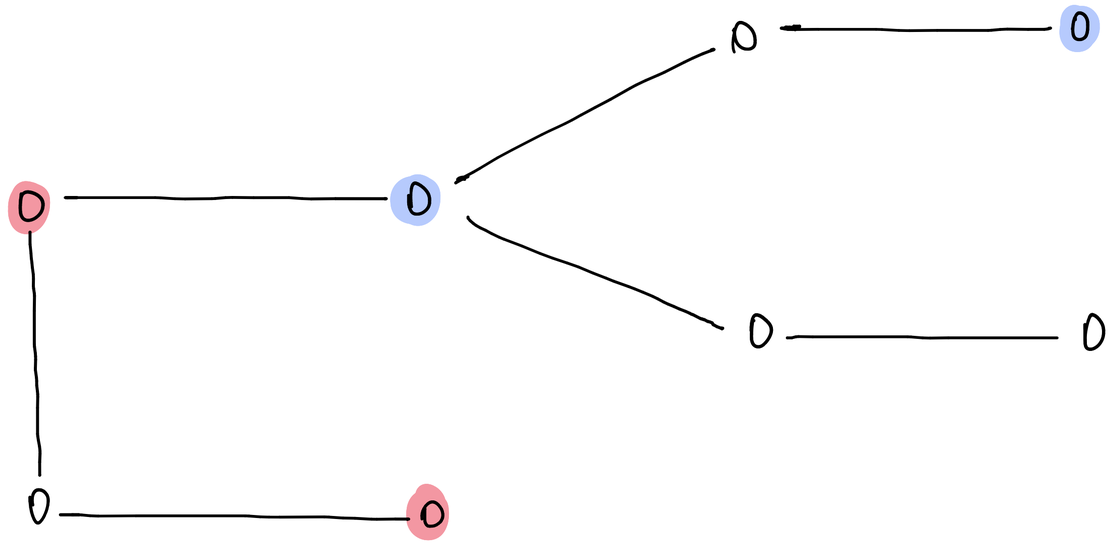


CoL $\leq$ NoGo



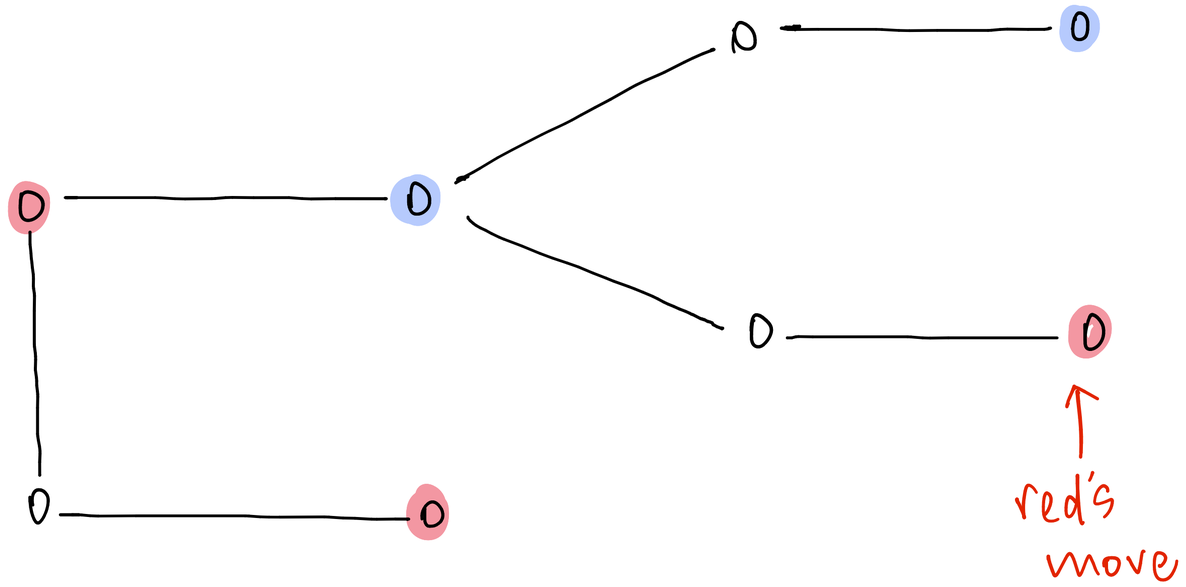
Col $\leq$ NoGo

Col: Cannot play at vertices adjacent to your own color.

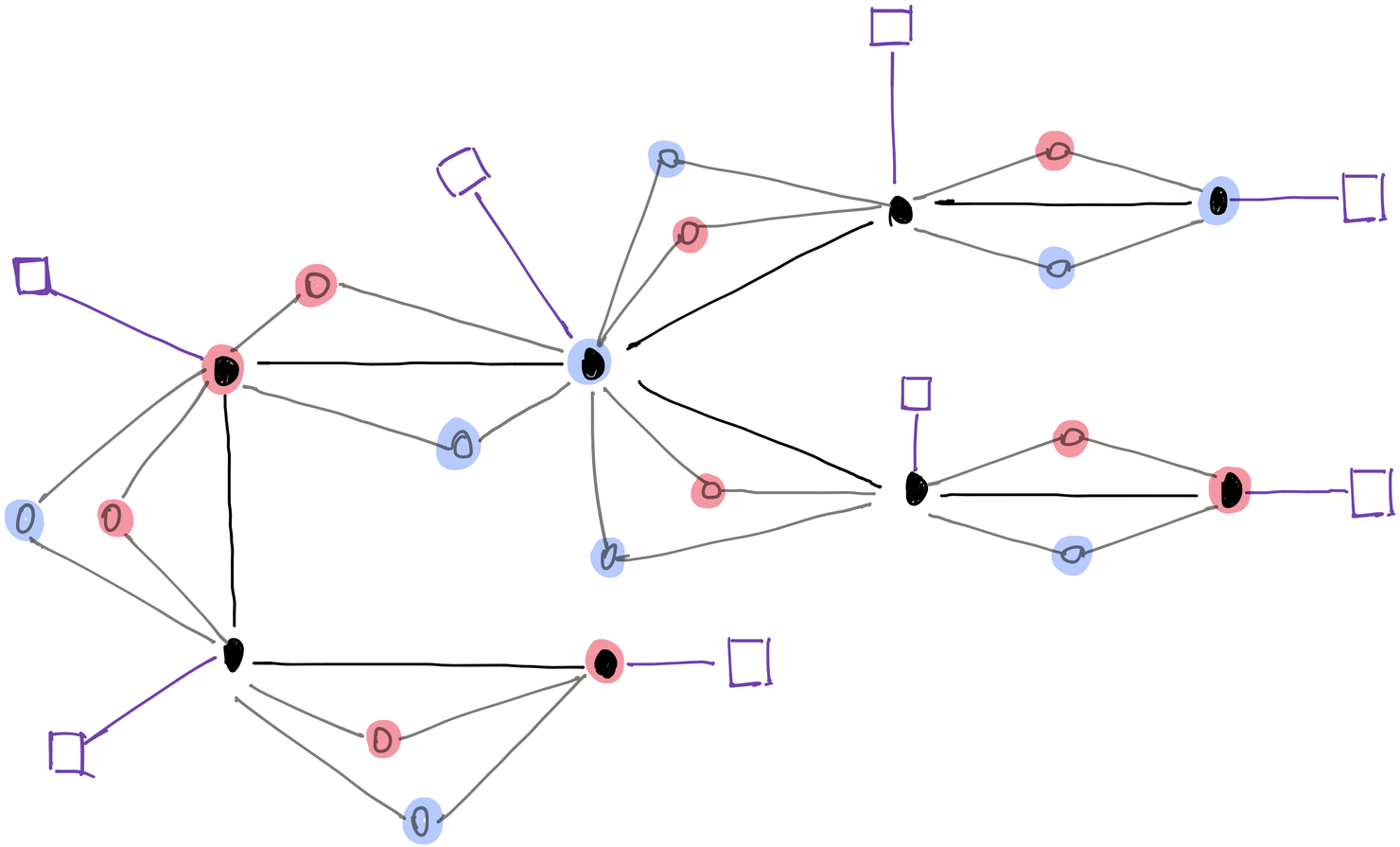


CoL $\leq$ NoGo

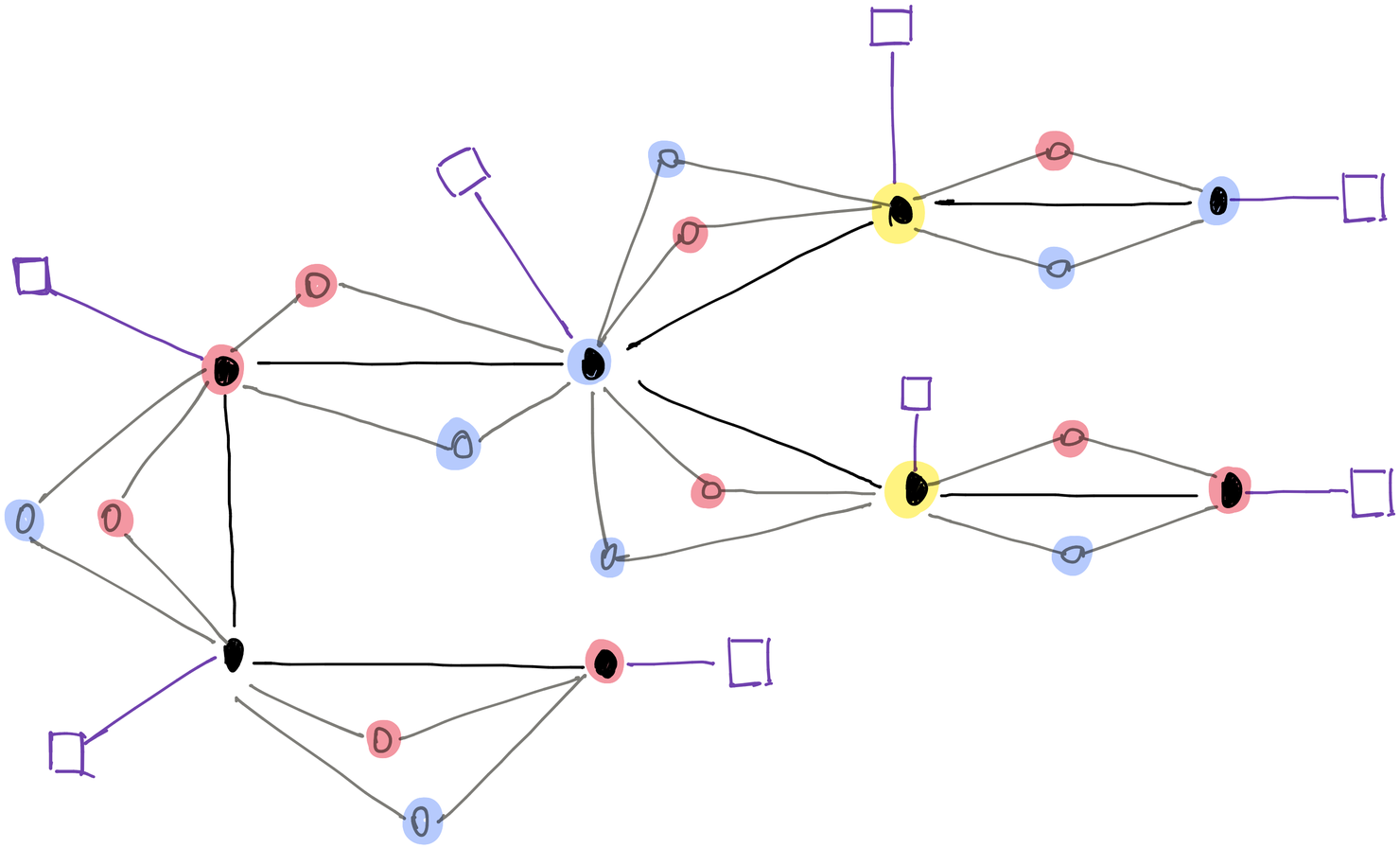
CoL: Cannot play at vertices adjacent to your own color.



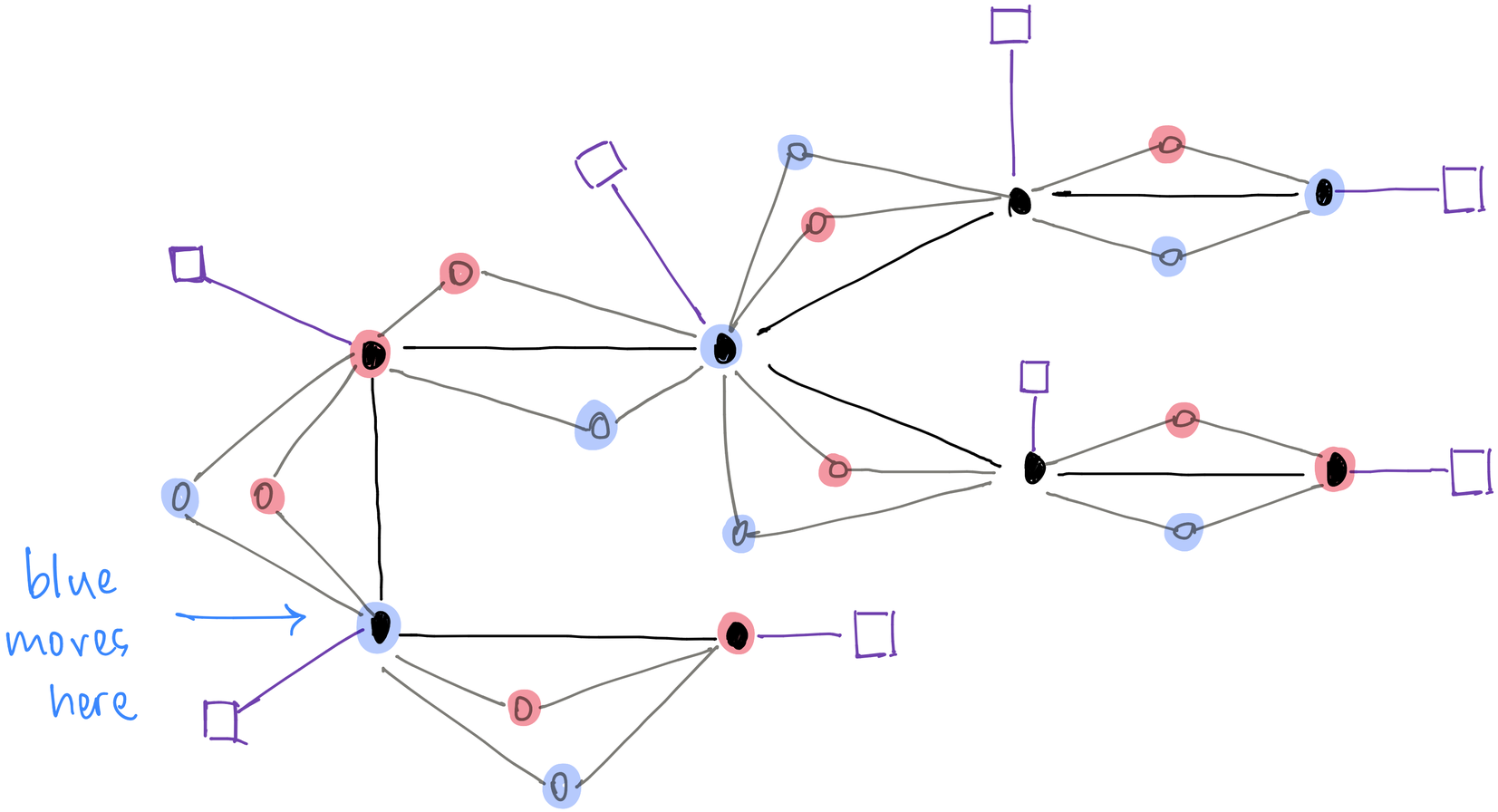
$\mathrm{Col} \leq \mathrm{NoGo}$



Col $\leq$ NoGo

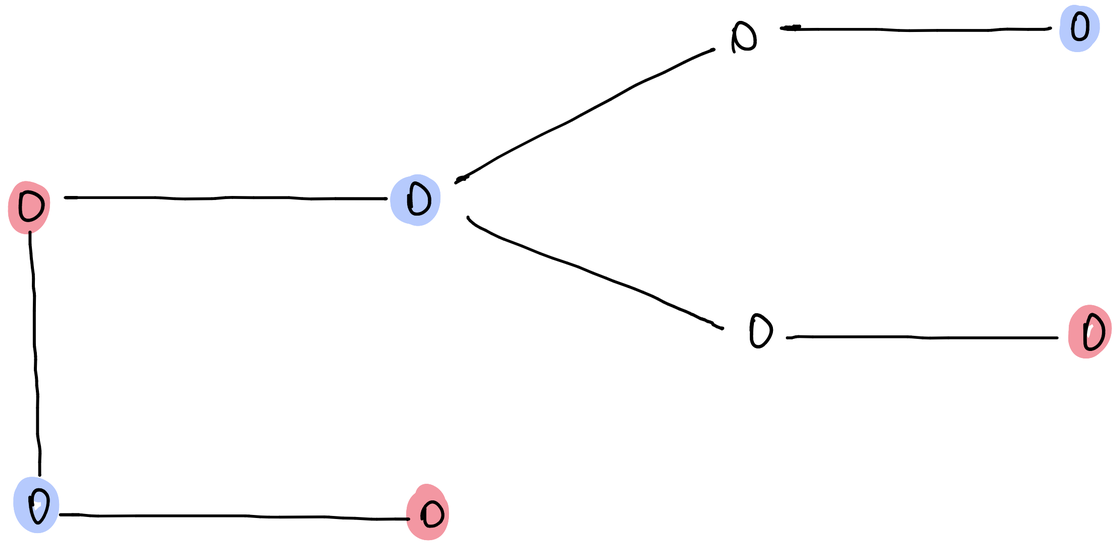


CoL ≤ NoGo



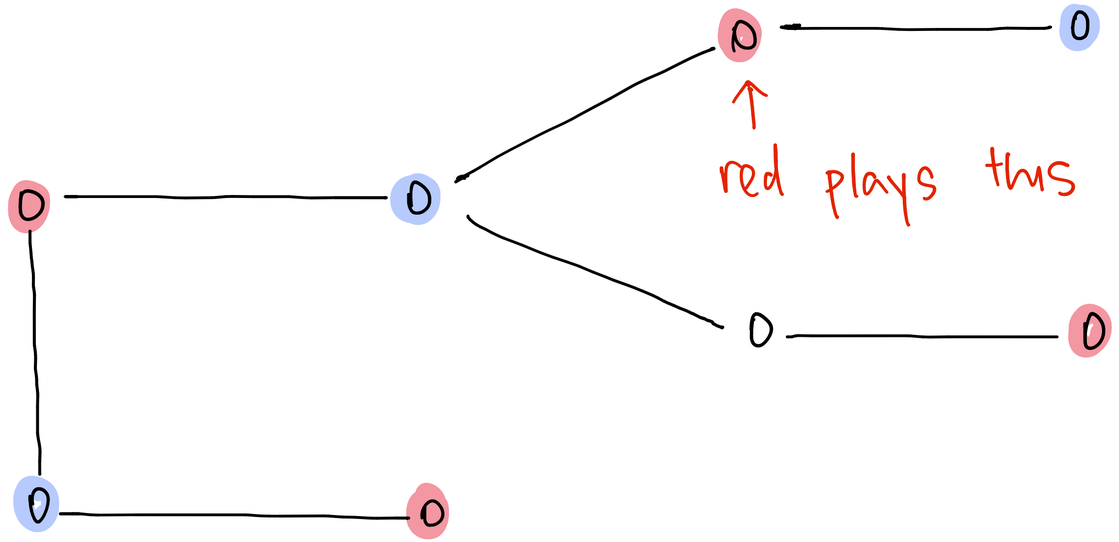
Col $\leq$ NoGo

Col: Cannot play at vertices adjacent to your own color.

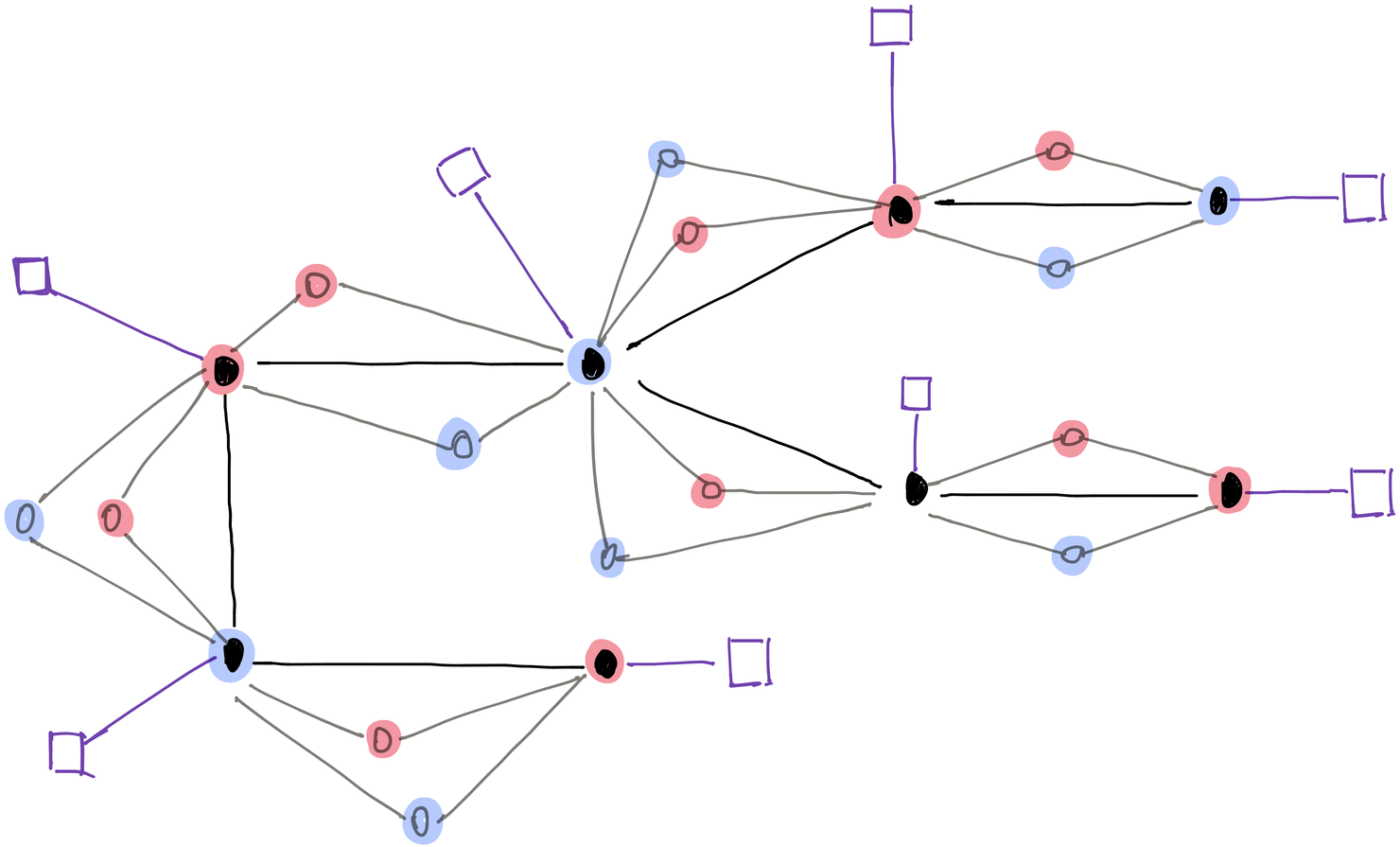


CoL $\leq$ NoGo

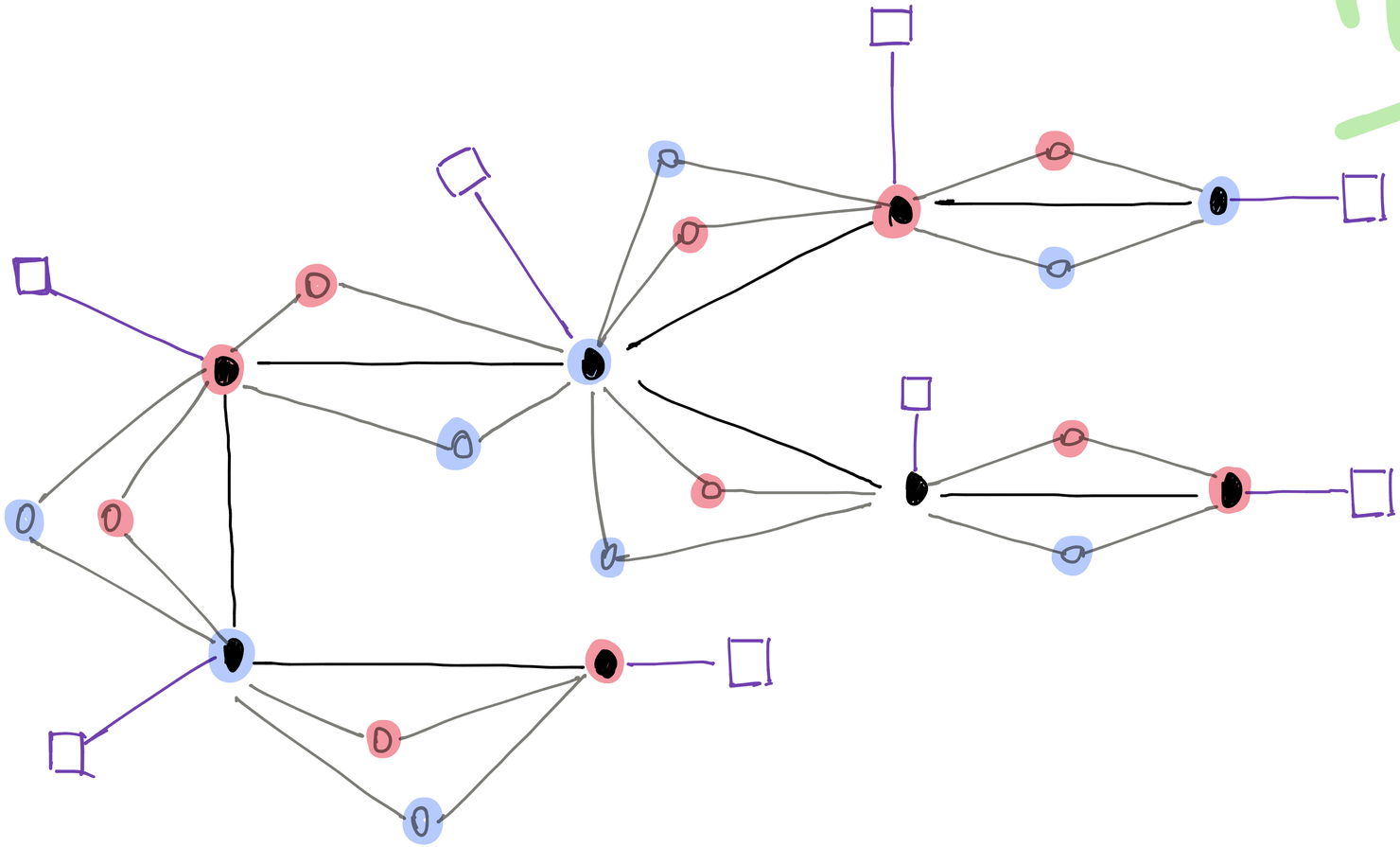
CoL: Cannot play at vertices adjacent to your own color.



$\mathrm{Col} \leq \mathrm{NoGo}$



Col $\leq$ NoGo



Summary

We transform a CoL board

to an equivalent NoGo board.

Summary

We transform a CoL board

to an equivalent NoGo board.

NoGo is as hard as CoL.

Summary

We transform a CoL board

to an equivalent NoGo board.

NoGo is at least as hard as CoL.

Geography

(Antakshari but w/ places)

Geography

(Antakshari but w/ places)

Gandhinagar

Geography

(Antakshari but w/ places)

Gandhinagar ⇝ Ropar

Geography

(Antakshari but w/ places)

Gandhinagar ⇝ Ropar ⇝ Raipur

Geography

(Antakshari but w/ places)

Gandhinagar ⇝ Ropar ⇝ Raipur ⇝ Roorkee

Geography

(Antakshari but w/ places)

Gandhinagar ⇝ Ropar ⇝ Raipur ⇝ Roorkee ⇝ Erode

Geography

(Antakshari but w/ places)

Gandhinagar ⇝ Ropar ⇝ Raipur ⇝ Roorkee ⇝ Erode

⇝ Élora

Geography

(Antakshari but w/ places)

Gandhinagar ⇝ Ropar ⇝ Raipur ⇝ Roorkee ⇝ Erode

⇝ Elora ⇝ Ahmedabad

Geography

(Antakshari but w/ places)

Gandhinagar ⇝ Ropar ⇝ Raipur ⇝ Roorkee ⇝ Erode

⇝ Elora ⇝ Ahmedabad ⇝ Delhi

Geography

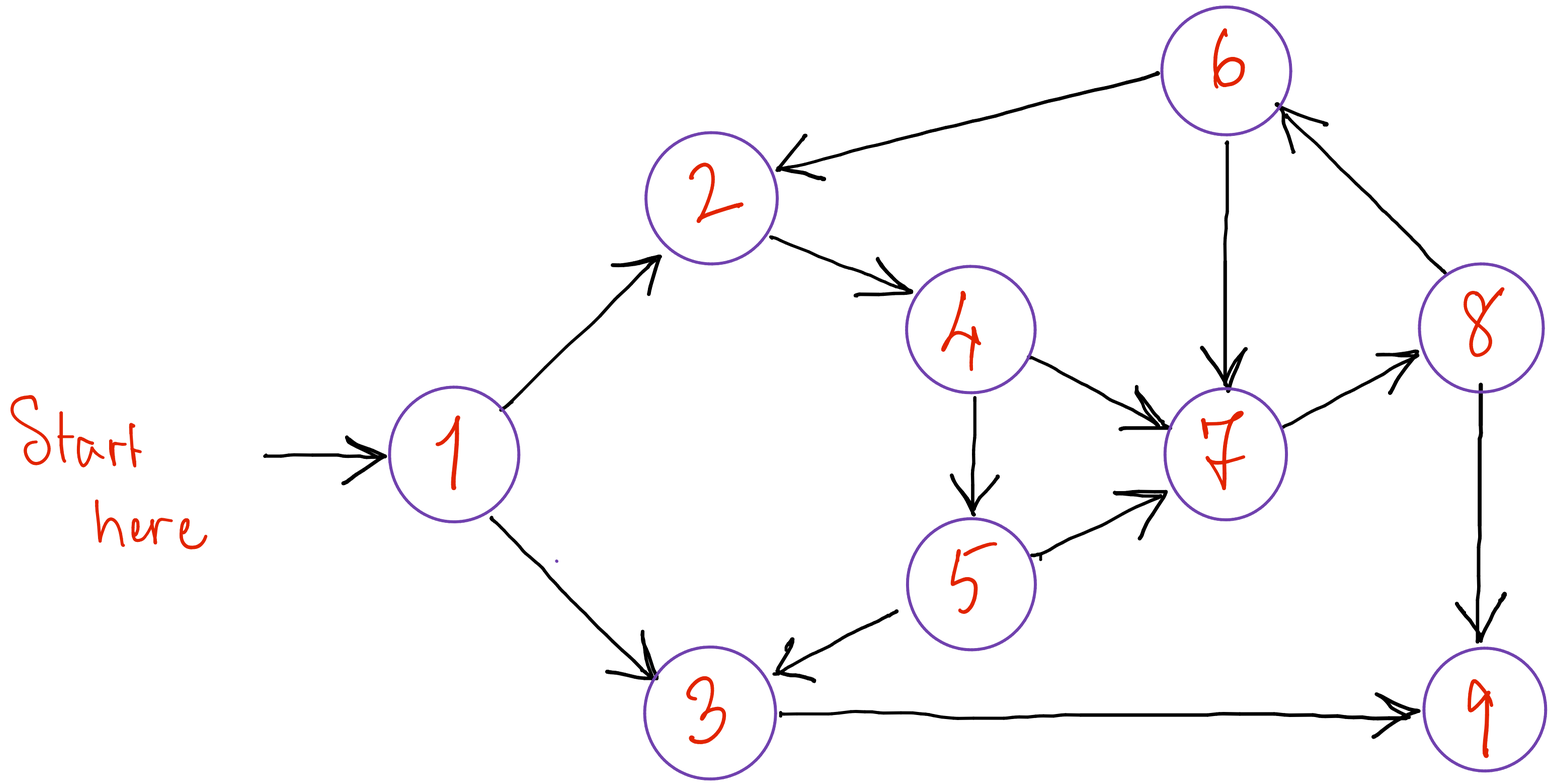
(Antakshari but w/ places)

Gandhinagar ⇝ Ropar ⇝ Raipur ⇝ Roorkee ⇝ Erode

⇝ Elora ⇝ Ahmedabad ⇝ Delhi ⇝ Indore

Generalized Geography

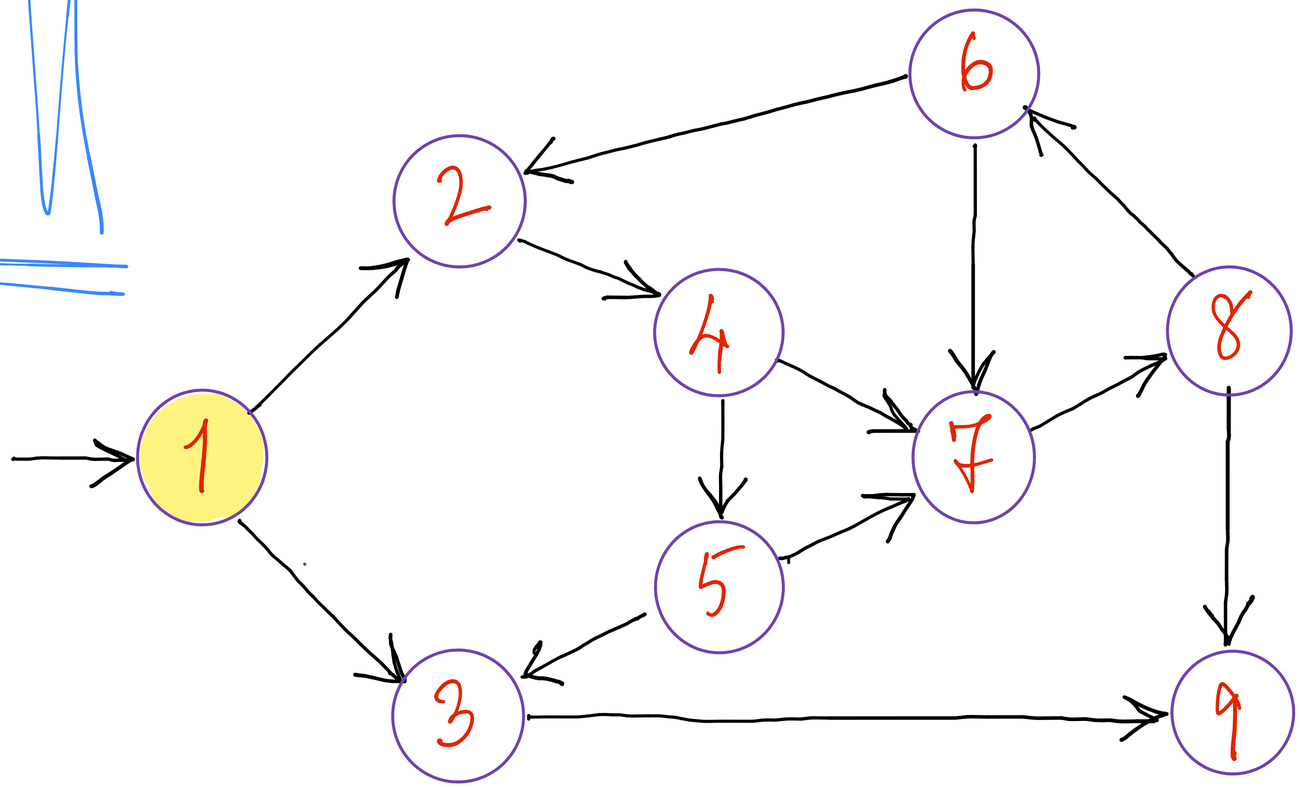
(played on an arbitrary directed graph)



Generalized Geography

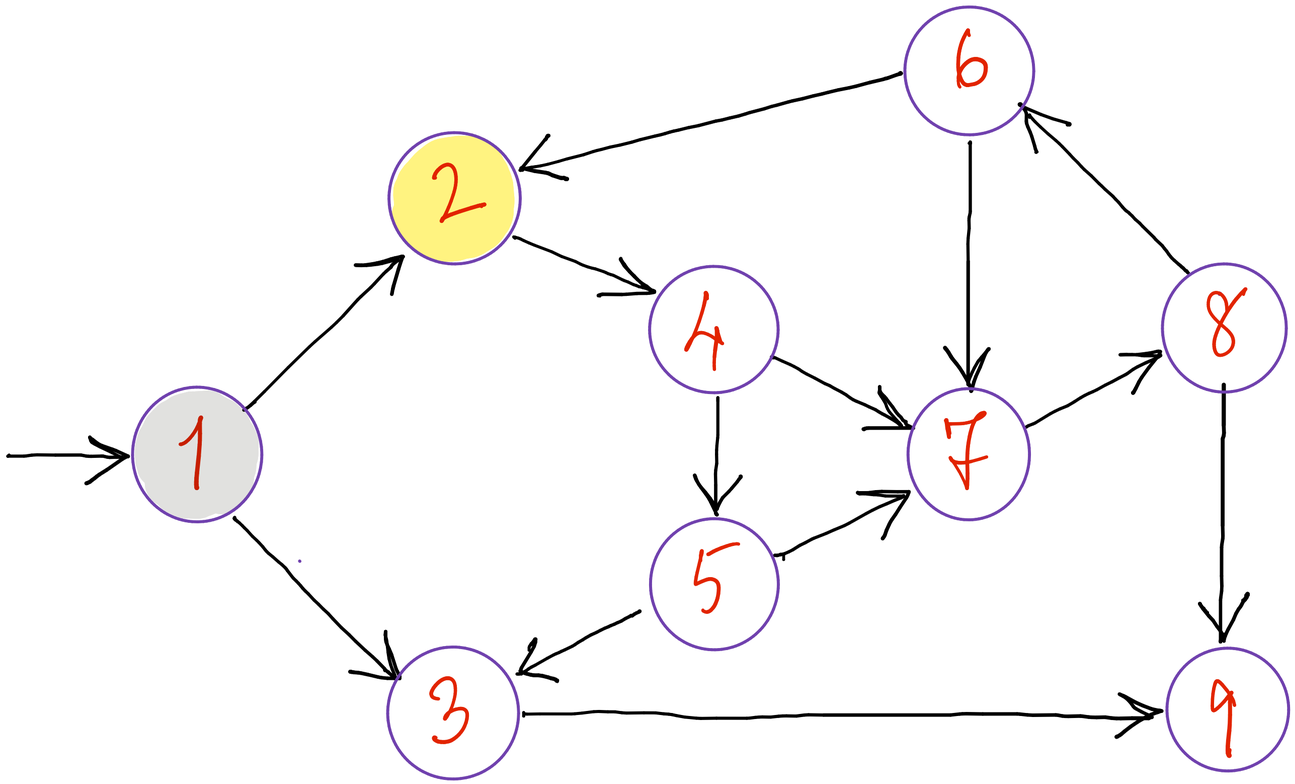
(played on an arbitrary directed graph)

Turn to move 1



Generalized Geography

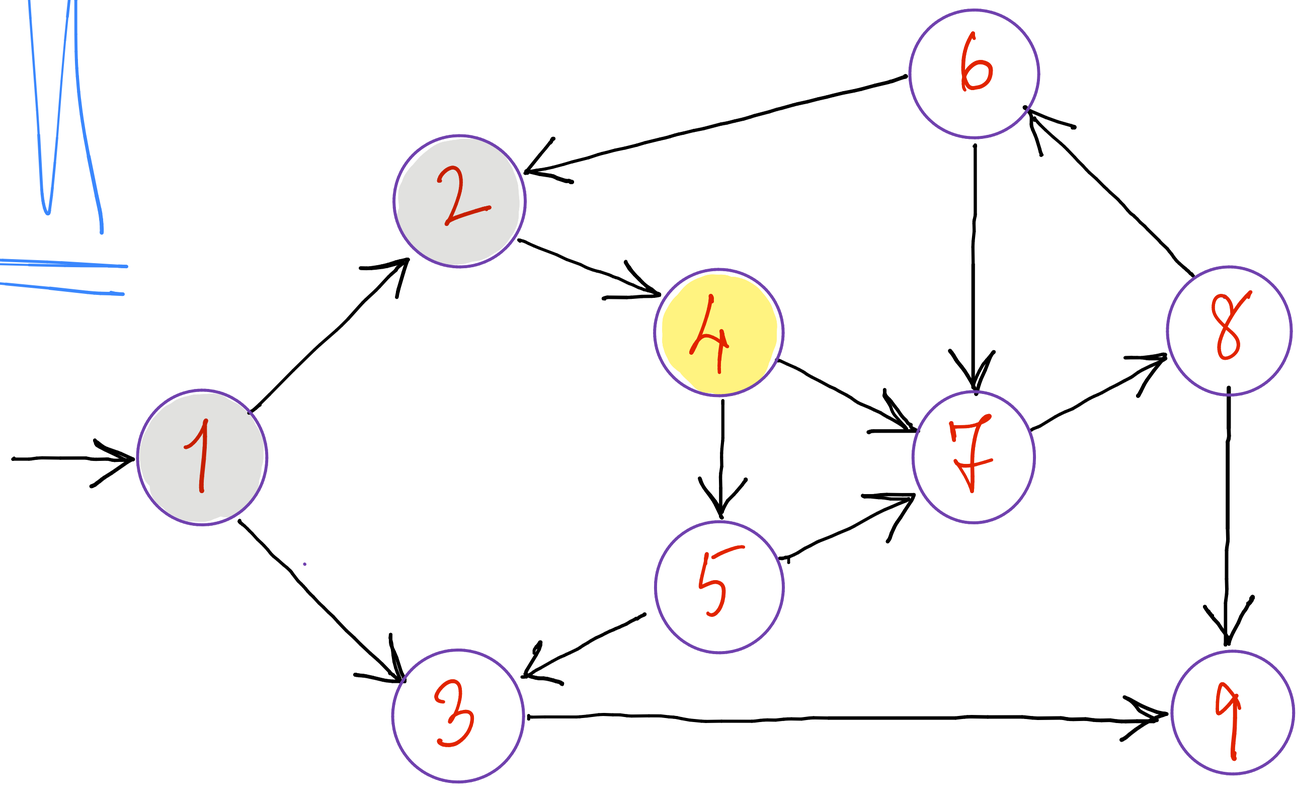
(played on an arbitrary directed graph)



Generalized Geography

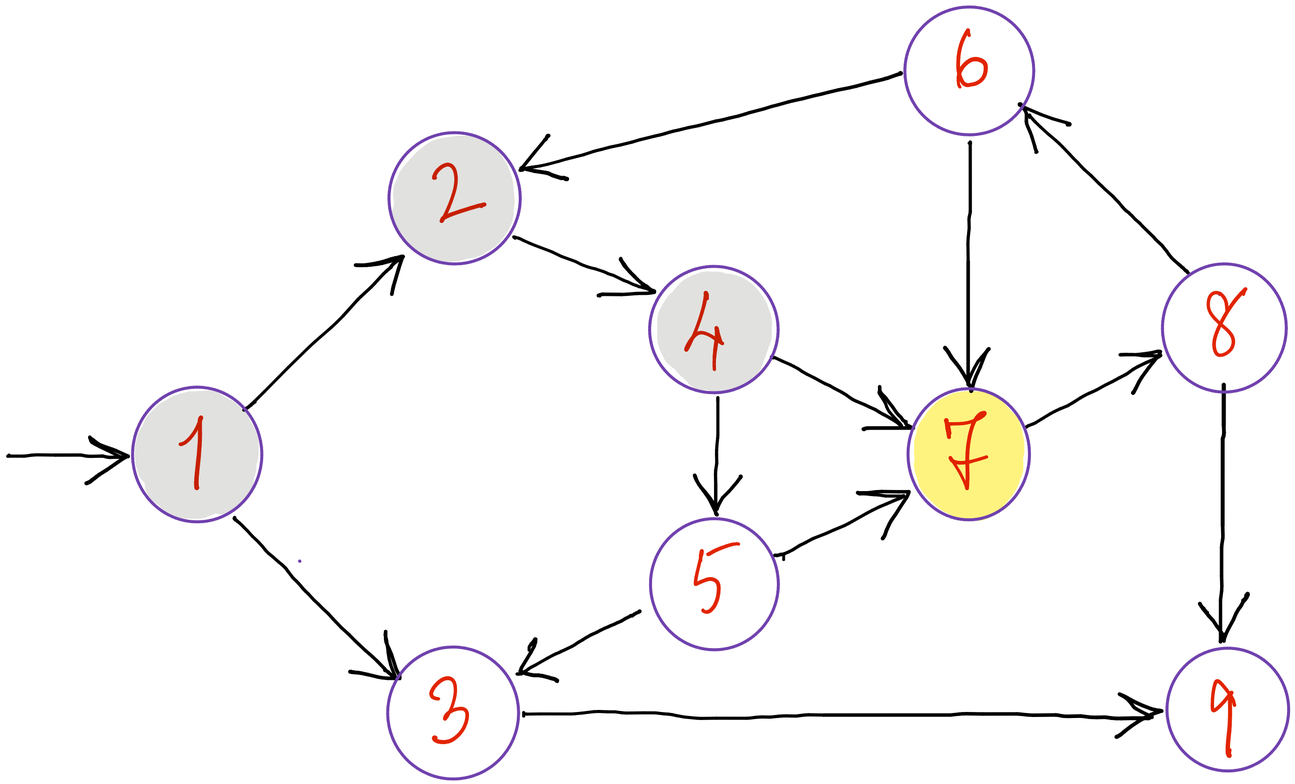
(played on an arbitrary directed graph)

Turn to move 1



Generalized Geography

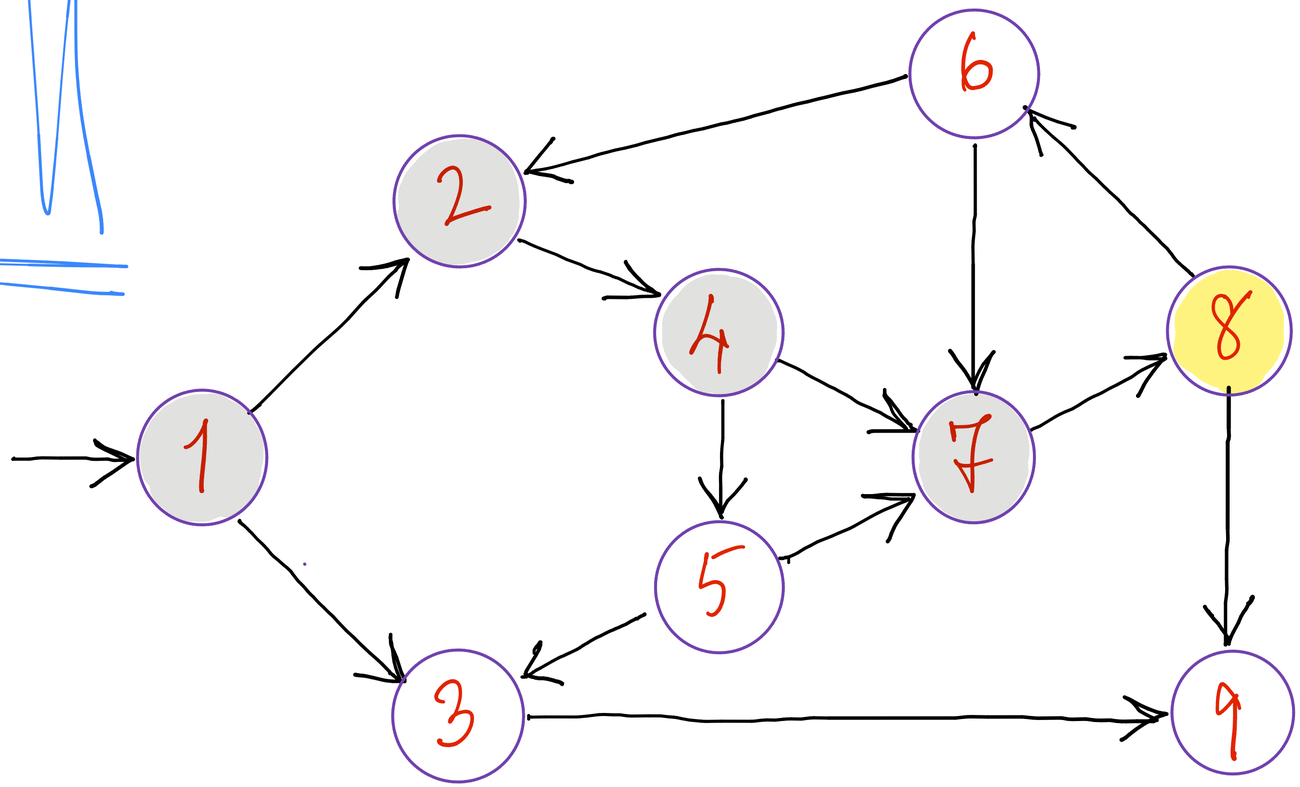
(played on an arbitrary directed graph)



Generalized Geography

(played on an arbitrary directed graph)

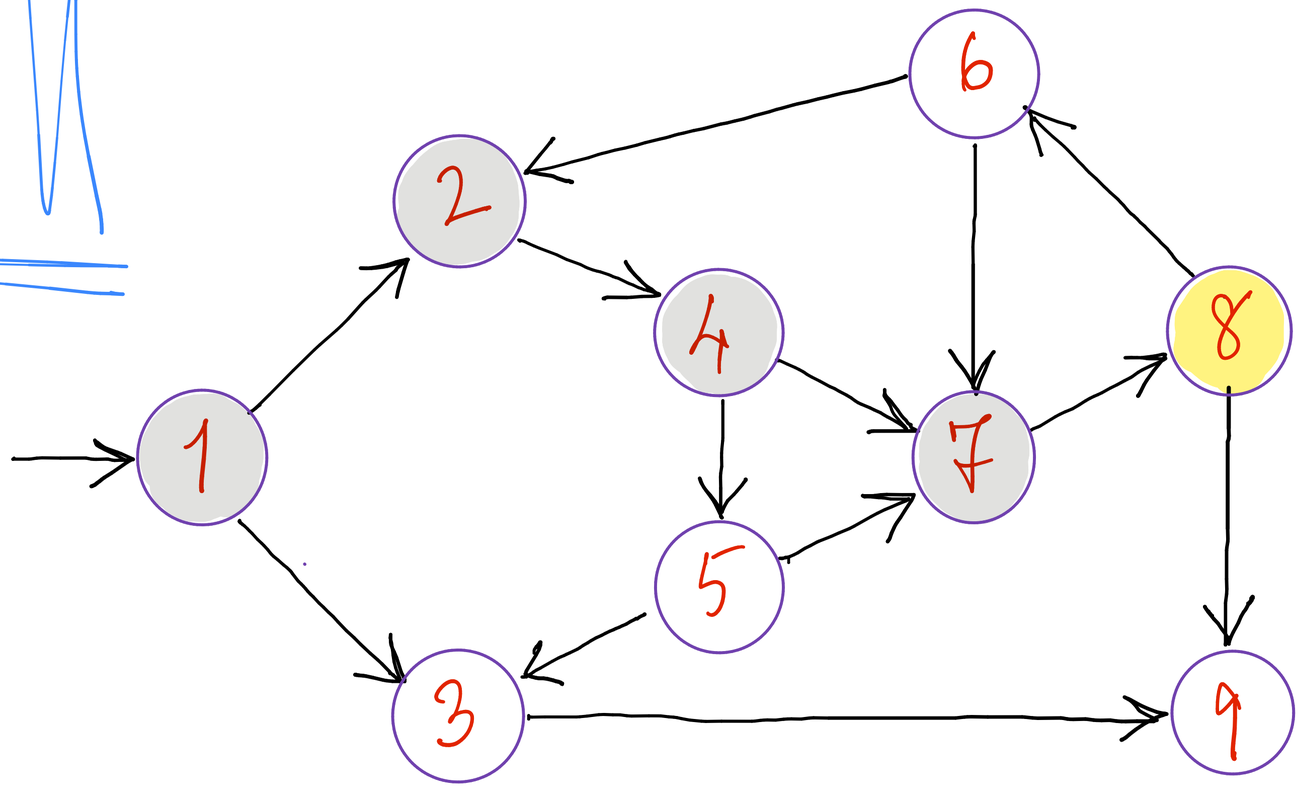
Turn to move 1



Generalized Geography

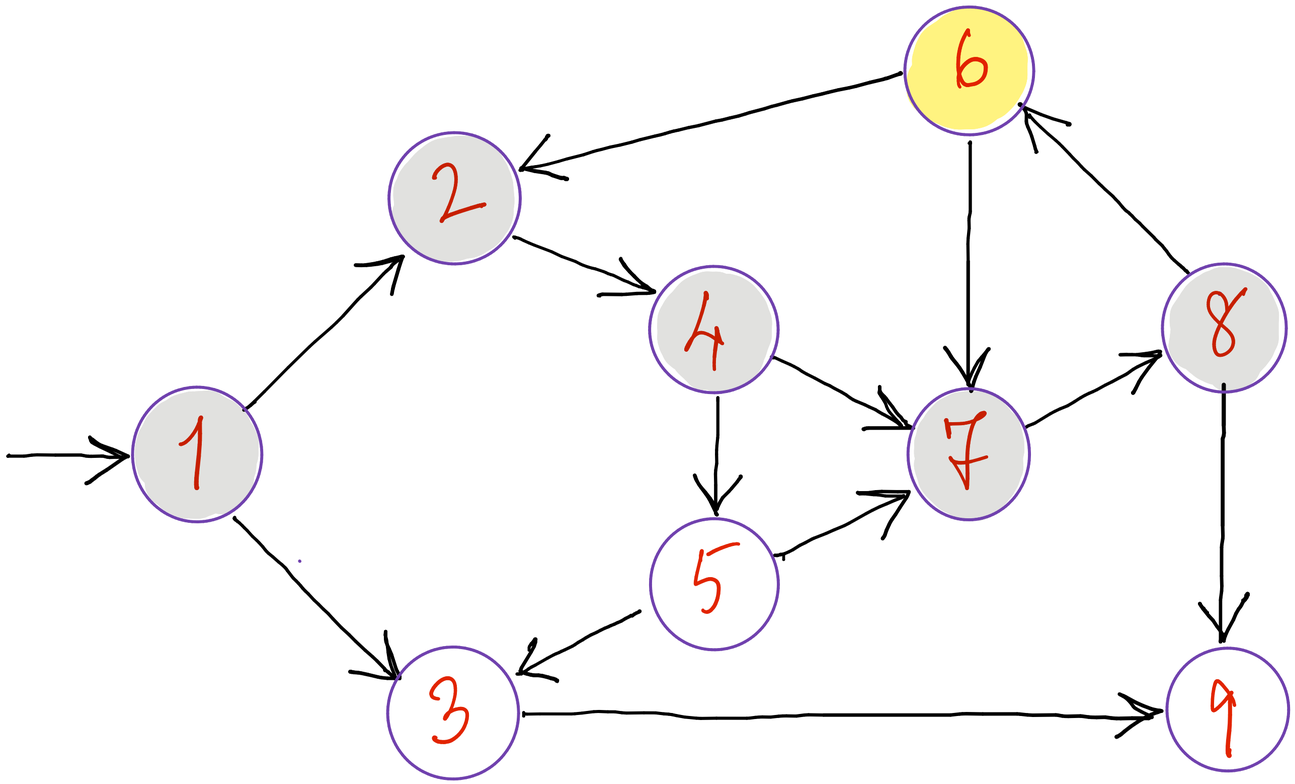
(played on an arbitrary directed graph)

Turn to move 1



Generalized Geography

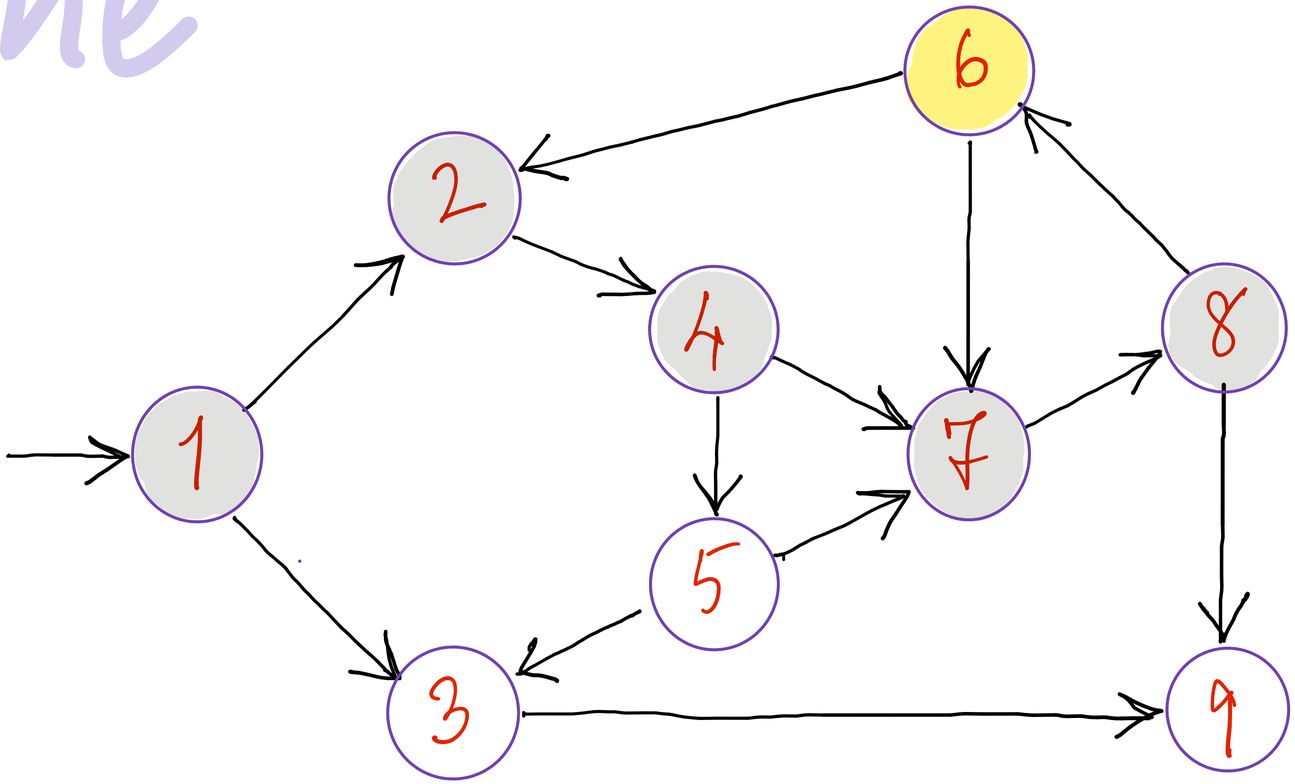
(played on an arbitrary directed graph)



Generalized Geography

(played on an arbitrary directed graph)

Game



Over

Player 1 wins

Boolean Formula Game

?

Boolean Formula Game

"Board" → A CNF formula over n boolean vars.

Boolean formula Game

"Board" → A CNF formula over n boolean vars.

$$(\bar{x}_0 \vee \bar{x}_2 \vee x_4) \wedge (x_0 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_4 \vee x_5)$$

Boolean formula Game

"Board" → A CNF formula over n boolean vars.

$$(\bar{x}_0 \vee \bar{x}_2 \vee x_4) \wedge (x_0 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_4 \vee x_5)$$

True player ⇝ wants to SATISFY the formula

Boolean formula Game

"Board" → A CNF formula over n boolean vars.

$$(\bar{x}_0 \vee \bar{x}_2 \vee x_4) \wedge (x_0 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_4 \vee x_5)$$

True player ⇝ wants to SATISFY the formula

false player ⇝ wants to FALSIFY the formula

Boolean formula Game

"Board" → A CNF formula over n boolean vars.

$$(\bar{x}_0 \vee \bar{x}_2 \vee x_4) \wedge (x_0 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_4 \vee x_5)$$

Turn based. $(2i-1)$-th round $\rightsquigarrow$ P1 sets x_{2i+1}

$(2i)$-th round $\rightsquigarrow$ P2 sets x_{2i}

Boolean formula Game

"Board" → A CNF formula over n boolean vars.

$$(\bar{x}_0 \vee \bar{x}_2 \vee x_4) \wedge (x_0 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_4 \vee x_5)$$

Boolean formula Game

"Board" → A CNF formula over n boolean vars.

$$(\bar{x}_0 \vee \bar{x}_2 \vee x_4) \wedge (x_0 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_4 \vee x_5)$$

Player 1. $\tau(x_0) = \text{true}$

Boolean formula Game

"Board" → A CNF formula over n boolean vars.

$$(\bar{x}_0 \vee \bar{x}_2 \vee x_4) \wedge (x_0 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_4 \vee x_5)$$

Player 1. $\tau(x_0) = \text{true}$

Boolean formula Game

"Board" → A CNF formula over n boolean vars.

$$(\bar{x}_0 \vee \bar{x}_2 \vee x_4) \wedge (x_0 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_4 \vee x_5)$$

Player 1. $\tau(x_0) = \text{true}$

Player 2. $\tau(x_1) = \text{false}$

Boolean formula Game

"Board" → A CNF formula over n boolean vars.

$$(\bar{x}_0 \vee \bar{x}_2 \vee x_4) \wedge (x_0 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_4 \vee x_5)$$

Player 1. $\tau(x_0) = \text{true}$

Player 2. $\tau(x_1) = \text{false}$

Boolean Formula Game

"Board" → A CNF formula over n boolean vars.

$$(\bar{x}_0 \vee \bar{x}_2 \vee x_4) \wedge (x_0 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_4 \vee x_5)$$

Player 1. $\tau(x_0) = \text{true}$

Player 2. $\tau(x_1) = \text{false}$

Player 1. $\tau(x_2) = \text{false}$

Boolean formula Game

"Board" → A CNF formula over n boolean vars.

$$(\bar{x}_0 \vee \bar{x}_2 \vee x_4) \wedge (x_0 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_4 \vee x_5)$$

Player 1. $\tau(x_0) = \text{true}$

Player 2. $\tau(x_1) = \text{false}$

Player 1. $\tau(x_2) = \text{false}$

Boolean formula Game

"Board" → A CNF formula over n boolean vars.

$$(\bar{x}_0 \vee \bar{x}_2 \vee x_4) \wedge (x_0 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_1 \vee \bar{x}_4 \vee \bar{x}_5) \wedge (x_1 \vee \bar{x}_4 \vee x_5)$$

Player 1. $\tau(x_0) = \text{true}$

Player 2. $\tau(x_1) = \text{false}$

Player 1. $\tau(x_2) = \text{false}$

Player 1 WIN

Boolean Formula Game $\rightsquigarrow$ Generalized Geography

Goal. Transform a formula

into a graph!

(while preserving equivalence)

Boolean Formula Game $\rightsquigarrow$ Generalized Geography