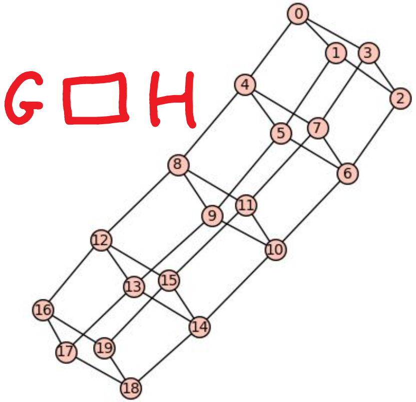
Two Graph Products (Cartesian and Strong) and the Cops and Robber Game

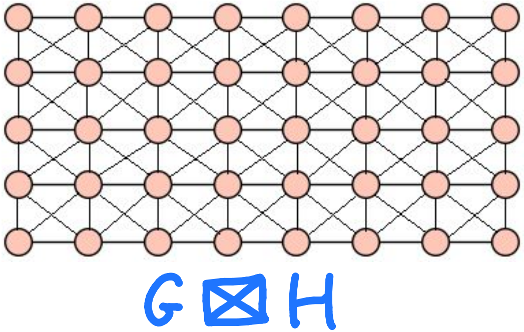


GRASCan August 2021

Brendan W. Sullivan
Emmanuel College, Boston

sullivanb@emmanuel.edu

@professorbrenda



Abstract

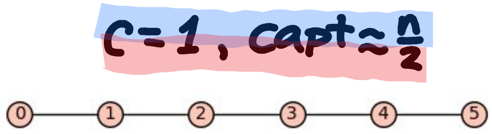
This talk will share a few related ideas about graph products and the cops and robber game. First, I will share some results and conjectures about how the cop numbers of G and H relate to the cop numbers of $G \square H$ (**Cartesian** product) and $G \boxtimes H$ (**Strong** product), for both the **ordinary** (all cops can move per turn) and **lazy** (only one cop can move per turn) variants. Second, I will describe how these two graph products play starring roles in some SageMath code that student researchers and I have used to calculate cop numbers and test conjectures.

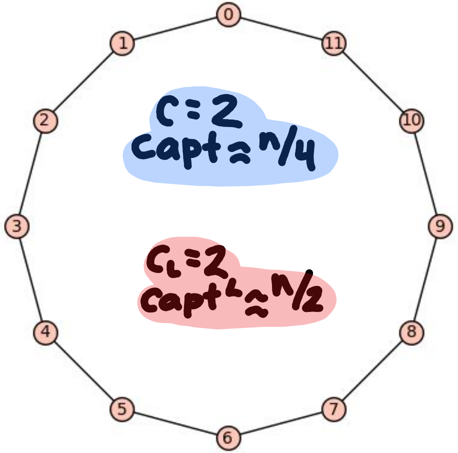
(The code is based on an algorithm described in: N. Clarke and G. MacGillivray, Characterizations of k-copwin graphs, *Discrete Math.*, **312** (2012) 1421-1425).

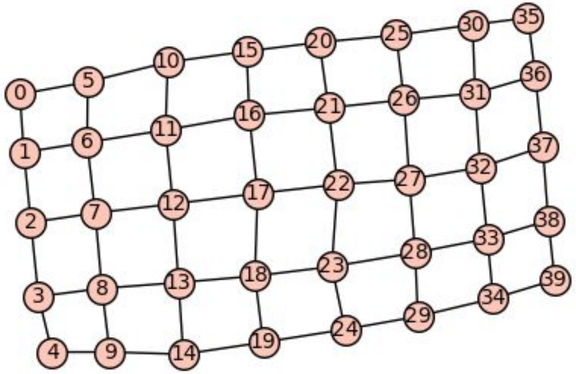
Joint work with Sean McGovern and past undergraduate research students: Niko Townsend, Mikayla Werzanski, Sarai Dancy, Bo McCormack, Osarumen Edosomwan

Introduction: Cops and Robber Game

1. Cop(s) initialize anywhere. Cops may occupy same vertex. Robber initializes in response.
2. Everyone knows where everyone is.
3. Cops get to move. Robber responds.
   a. Legal move: along an edge
   b. Ordinary game: *all* Cops may move
   c. "Lazy" variant: *only one* Cop may move
   d. In both: *pass (no move)* is a legal option. (**Active game** forces both sides to make a move each turn.)
4. If *there exists a strategy* whereby a Cop lands on the Robber, then the Cop squad wins.
5. Otherwise, there exists a strategy whereby the Robber evades capture forever, so Robber "wins".



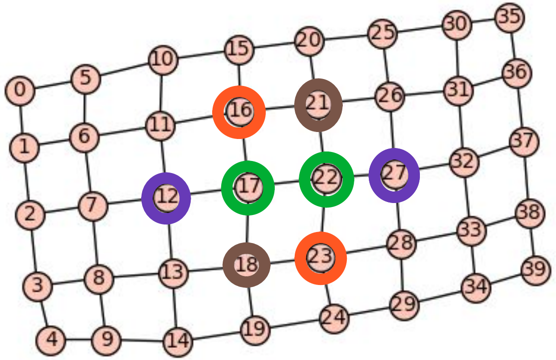




Graph Parameters: **Cop Number** and **Capture Time**

1. The **Cop Number** *$c(G)$* of a graph is the minimum k such that k Cops have a winning strategy, yet the Robber has a winning strategy against k-1 Cops.
   a. Must consider both perspectives. In practice, Robber strategies seem challenging and rarer.
   b. Analogously, define the **Lazy Cop Number** $c_L(G)$ of a graph (only 1 Cop per turn).
   c. The Cop Number is the *minimum* such k. Still reasonable to play with more than that many.
2. Given G and any $k \geq c(G)$, the **Capture Time** $capt_k(G)$ is the number of moves it will take for k Cops to win, assuming optimal play on both sides.
   a. Consider both perspectives: Cops want to win quickly, Robber wants to prolong the inevitable.
   b. In general, there seem to be fewer results about Capture Times than Cop Numbers, and even fewer for the **overprescribed** game (where k is strictly more than the minimum necessary), as well as for the Lazy variant $capt_k^L(G)$.

Example: Cop Number & Capture Time



Outputs from running code:

- True, 2 ordinary Cops win
- Capture Time is 5 moves
- List of starting locations for Cops to achieve that optimal Capture Time: [(12,27), (16,23), (17,22), (18,21)]
- Stores a *strategy dictionary*: given Cops' location and where Robber is about to move, states optimal move for Cops
- Similarly, given location of all players, states how many moves from capture
- Can test *Lazy* Cops, as well (often faster)

```
Yes,  2  ordinary chasers win on G! Capture time is  5
Start the chasers at any of  4  positions to achieve capture time.
Ending elapsed time:  3.5229885578155518
```

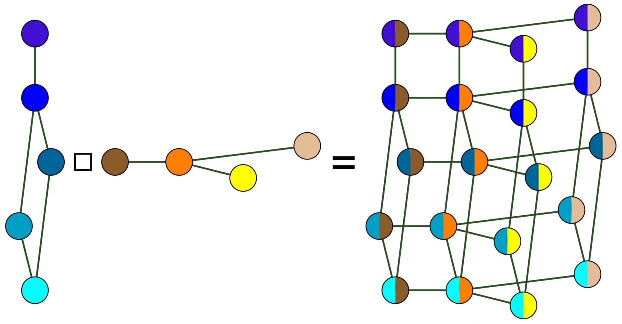
```
Yes,  2  lazy chasers win on G! Capture time is  9
Start the chasers at any of  12  positions to achieve capture time.
Ending elapsed time:  3.0486719608306885
```

Known: capture time of $m \times n$ grid is $\lfloor \frac{m+n}{2} \rfloor - 1$ (Mehrabian, *Disc. Math.* 2010)

Open: Higher dimensional grids?
Conjecture: Capture time for *Lazy*: *m+n-4*

Graph Products: **Cartesian & Strong**

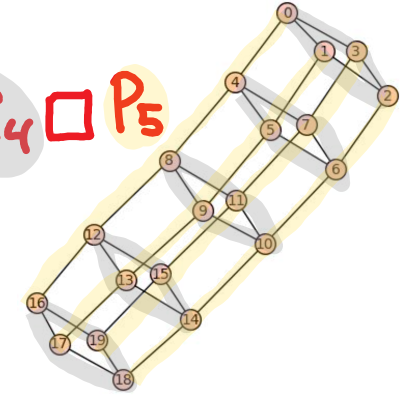
Wikipedia: Cartesian product of graphs



Any graph product *G “times” H* yields a graph whose vertex set is ordered pairs (vtx of *G*, vtx of *H*). The different products are determined by which edges get included.

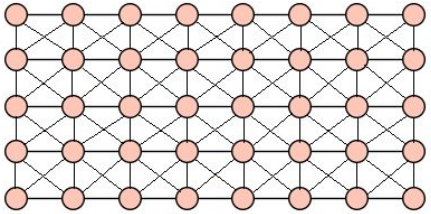
Cartesian Product $\square$: (a,b)~(a,d) or (a,b)~(c,b), can change *only one coordinate at a time*

$C_4 \square P_5$



Strong Product $\boxtimes$: *also* includes (a,b)~(c,d), can change *any number of coordinates at a time*

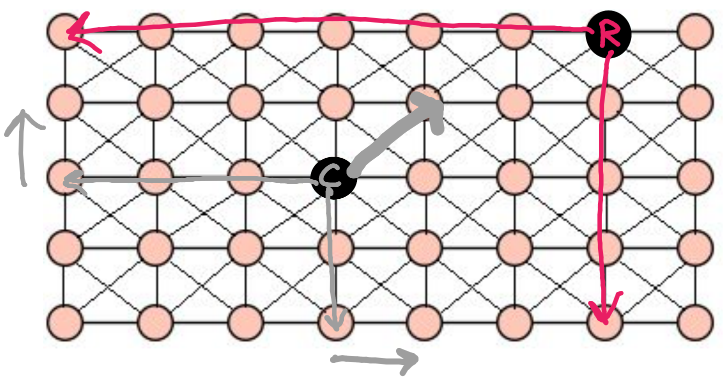
$P_5 \boxtimes P_8$



Playing on graphs of the form $G \square H$ or $G \boxtimes H$ can be broken down into strategies on the factors *G*,*H*

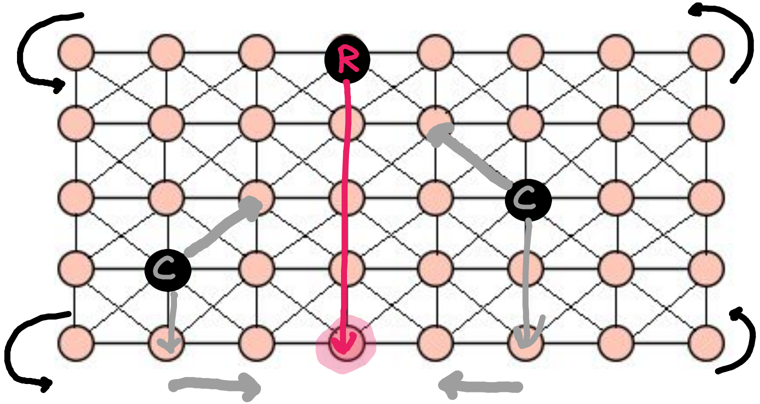
Example: Enacting Strategies on Factors in Graph Product

Consider a *lattice grid*: $P_5 \boxtimes P_8$



- Optimal strategy for Cop on each path factor: start in middle and advance toward Robber.
- With *strong product*, can enact both strategies at once: change *both* coordinates.

Likewise, with a *cylindrical grid:* $P_5 \boxtimes C_8$



- Two Cops play against Robber's *shadow* on the cycle factor, regardless of where the Robber really is along the path factor.
- Simultaneously play strategy on path factor.

Example Result: Cop Number of Strong Product Graph

Theorem: $c(G \boxtimes H) = c(G) + c(H) - 1$

Proof. Must exhibit (1) a Cop strategy with *c(G)+c(H)-1* many Cops, as well as (2) a Robber strategy against *c(G)+c(H)-2* many Cops.

1. See: S. Neufeld and R. Nowakowski, A game of cops and robbers played on products of graphs, *Discrete Math.*, **186** (1998) 253-268. **Main idea:** play against Robber's shadow on *G* until enough Cops are shadowing that vertex. Thereafter, they keep shadowing that vertex while enacting the winning strategy on *H* until capture.

2. See: B. Sullivan, "Lazy Cops and Robbers on Product Graphs", *JMM 2016*. **Main idea:** Robber divides cops into *(c(G)-1)+(c(H)-1)* a priori, simultaneously enacts winning strategy on *G* against first squad and winning strategy on *H* against second squad.

Known Results: Cop Numbers of Graph Products

1. Tosic 1987: upper bound for Cartesian product $c(G \square H) \leq c(G) + c(H)$
2. Maamoun & Meyniel 1987: Cartesian product of trees $c(T_1 \square \cdots T_j) = \left\lceil \frac{j+1}{2} \right\rceil$
3. Neufeld & Nowakowski 1998:
   a. Cartesian product of cycles $c(\underbrace{C_1 \square \cdots \square C_k}) = k+1$
   b. Cartesian product of cycles and trees $c(\underset{G}{\downarrow} \square \underbrace{(T_1 \square \cdots \square T_j)}_{H}) = k + \left\lceil \frac{j+1}{2} \right\rceil$
   c. Partial characterization of when $c(G \square H) = c(G) + c(H) - 1$
4. Mehrabian 2011: capture time of Cartesian product of two trees $capt(T_1 \square T_2) = \left\lfloor \frac{1}{2} diam(T_1 \square T_2) \right\rfloor$

New Results: Cop Numbers of Graph Products

With student researchers, have analyzed some examples of *Ordinary* & *Lazy* game:

$c(G)+c(H)-1$

1. Cylindrical grids: $c_L(P\square C) = c(P\square C) = 2$
2. Torus grids: $c_L(C_n \square C_n) \le 2\cdot\lceil n/3 \rceil$, $c_L(C_n \boxtimes C_n) \le 2\cdot\lceil n/4 \rceil$,
3. “Jungle”: $c_L(K\square T) = c = 2$
4. “Ring portals”: $c(K\square C) = c_L = 3$, $c(K\boxtimes C) = c_L = 2$

$c(G)+c(H)$

In general, looking for patterns in results and how to prove winning strategies.

Conjectures: Cop Numbers of Graph Products

Based on those patterns in results and winning strategies, we have some ideas about graph products, in general:

1. Strong product theorem: *equality* could be helpful?

$$c(G \boxtimes H) = c(G) + c(H) - 1$$

2. Cartesian products:

$$\cancel{c(G \boxtimes H) \overset{?}{\leq} c(G \,\square\, H)}$$

$$c(G \,\square\, H) = c(G) + c(H) - \{0, 1, 2\}$$

"nearly c^* win"

[Neufeld + Nowakowski]

3. **Recently, a counterexample:**

$$c(\text{Petersen} \,\square\, C) = 3 + 2 - 2 = 3 \quad !?$$

minimal $c = 3$ $(c = c_L = \gamma)$

Hahaha I love that the smallest example involves the Petersen graph. GOAT counterexample

Jul 17, 2021, 10:12 PM

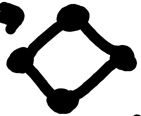
Conjectures: Cop Numbers of Graph Products

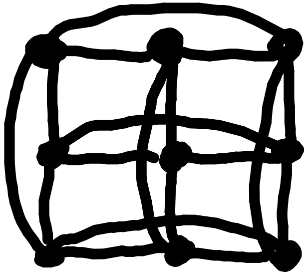
4. Adapting results/ideas to *Lazy* game:

$c_L(G \square H) \le c_L(G) + c_L(H) - 1$ ("G-squad")

$c_L(G \boxtimes H)$???

5. There may be a connection between graph products and **minimal examples** of graphs with certain Cop parameters.

   a. Smallest $c=c_L=2$:  C_4

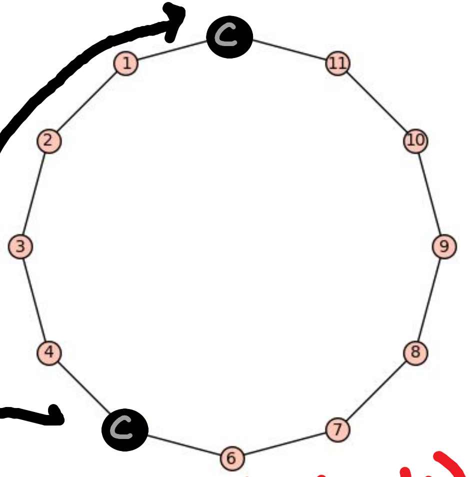
   b. Smallest $c_L=3$: $K_3 \square K_3 =$ 

   Sullivan, Townsend, Werzanski, The 3x3 rooks graph is the unique smallest graph with lazy cop number 3, *arXiv preprint*, arXiv:1606.08485 (2016)

Graph Products and the **Game State Graph**

Location of all k cops $\longleftrightarrow$ k-tuple of vertices of G

- **Ordinary:** each cop may move on a turn: each coordinate of tuple can change $\rightarrow$
- **Lazy:** only one cop may move per turn: only one coordinate can change at a time $\rightarrow$



$(0,5) \rightarrow (1,5),(0,4)$
$(11,5),(0,6)$

also
$(1,4),(1,6)$
$(11,4),(11,6)$

Exponential product graph G^k represents all possible locations (vertices of G^k) and moves (edges of G^k) that k cops can make while playing on the graph G.

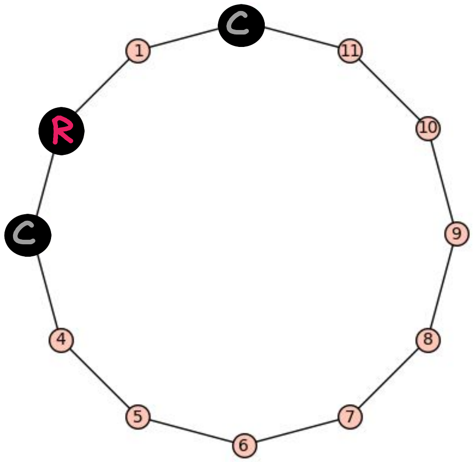
- Cartesian product for the Lazy game $G \square \cdots \square G$
- Strong product for the Ordinary game

$G \boxtimes \cdots \boxtimes G$

N. Clarke, G. MacGillivray, Characterizations of k-copwin graphs, *Discr. Math.* **312** (2012) 1421-1425.

Clarke & MacGillivray Algorithm

- Given G and k, form G^k
- Goal: each (v,p) gets a label $i \geq 0$ meaning, "If Robber is at vertex v in $V(G)$ and Cops' locations are given by p in $V(G^k)$, then Cops are at most i moves away from capture."
- Proceed inductively:
    - $i=0$ labels are capture states: e.g. (0, (0,3))
    - $i=1$ labels are "traps": for each possible Robber move, there is a capturing move in response
    - In general, (v,p) gets label i iff for all possible Robber moves $v \to w$, there exists a Cop response $p \to q$ such that (w,q) is labeled i-1 or less.



$(0,(0,3)) \to i=0$

$(2,(0,3)) \to i=1$

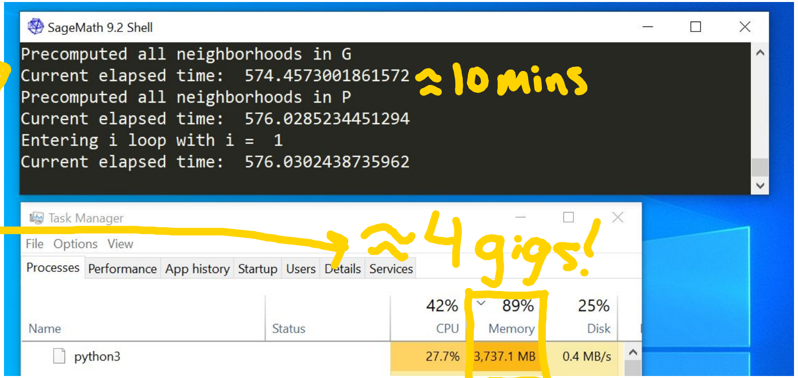
$(4,(0,5)) \to i=3$ Lazy

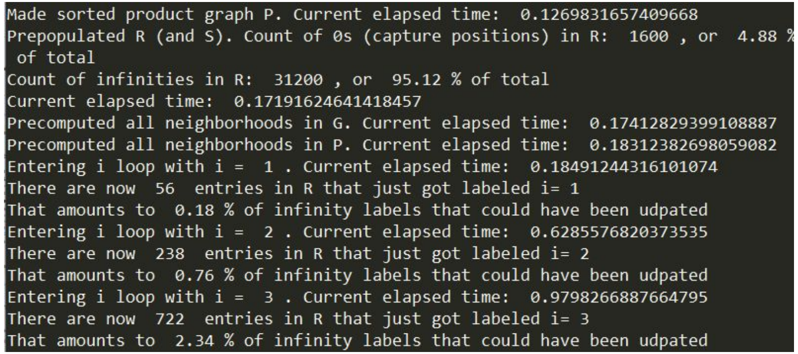
$i=2$ Ordinary

Implementation in Sage

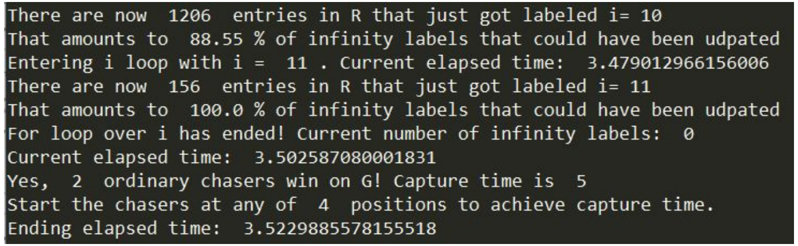
40 vtxs
k=3

- Subroutine to make exponential product graph: lots of time & memory!
  One speedup: cut out "redundant states"
- $O(n^{2k+2})$, where $n=|V(G)|$ and k Cops
- Loop over game states → *relational (R)* and *strategy (S)* dictionaries.
  - If any entry in R is "infinite", Cops lose
  - Else, Cops win and capture time is the minimax over possible starting locations
- Can store these R and S dictionaries afterwards as huge text files





```
Made sorted product graph P. Current elapsed time:  0.1269831657409668
Prepopulated R (and S). Count of 0s (capture positions) in R:  1600 , or  4.88 %
 of total
Count of infinities in R:  31200 , or  95.12 % of total
Current elapsed time:  0.17191624641418457
Precomputed all neighborhoods in G. Current elapsed time:  0.17412829399108887
Precomputed all neighborhoods in P. Current elapsed time:  0.18312382698059082
Entering i loop with i =  1 . Current elapsed time:  0.18491244316101074
There are now  56  entries in R that just got labeled i= 1
That amounts to  0.18 % of infinity labels that could have been udpated
Entering i loop with i =  2 . Current elapsed time:  0.6285576820373535
There are now  238  entries in R that just got labeled i= 2
That amounts to  0.76 % of infinity labels that could have been udpated
Entering i loop with i =  3 . Current elapsed time:  0.9798266887664795
There are now  722  entries in R that just got labeled i= 3
That amounts to  2.34 % of infinity labels that could have been udpated
```



```
There are now  1206  entries in R that just got labeled i= 10
That amounts to  88.55 % of infinity labels that could have been udpated
Entering i loop with i =  11 . Current elapsed time:  3.479012966156006
There are now  156  entries in R that just got labeled i= 11
That amounts to  100.0 % of infinity labels that could have been udpated
For loop over i has ended! Current number of infinity labels:  0
Current elapsed time:  3.502587080001831
Yes,  2  ordinary chasers win on G! Capture time is  5
Start the chasers at any of  4  positions to achieve capture time.
Ending elapsed time:  3.5229885578155518
```

Using Code to Test Conjectures

- Petersen times cycle broke a longstanding idea about $\cancel{c(G \square H) \overset{?}{\leq} c(G \square H)}$

  Never would have found that example by hand!

  $c(\text{Petersen} \square C) = 3 + 2 - 2 = 3$ $10 \times 4 = 40$ vtxs, $k = 3$

- Knowing $c = c_L = 2$ for rectangular and cylindrical grids, can calculate capture times and analyze code output files to determine optimal strategies.

  $\text{capt}_2^L(P_m \square P_n) = m + n - 4$

- Can run code on all graphs of certain size or with certain properties, hunting for examples.

all cop #s on $n = 10$ vtxs, $\simeq 12$ mil graphs, $\simeq 56$ hours

```
sage: c_star_10 = find_c_star_win_graphs(10)
All done, tested 11716571 graphs total.
Total runtime (seconds): 202080.8203523159
```

Future Work

1. Code takes a lot of time & memory! Can we be more efficient about how to make the exponential product graph, or what to do with it?
2. Use the code to calculate capture times and analyze optimal strategies, leading to provable results for Ordinary, Lazy, overprescribed, … more?
3. Could the *strategy dictionary* be turned into a playable AI for a game app?!
4. Hopefully make progress on conjectures about Cartesian & Strong products.
   a. Possibly characterize $c(G\square H) = c(G) + c(H) - \left\{\begin{matrix}0\\1\\2\end{matrix}\right\}$?
   b. Find examples and Improve bounds on $c_2(G\square H), c_L(G\boxtimes H)$
   c. And what can all of this teach us about the game itself or its applications?

References

1. N. Clarke and G. MacGillivray, Characterizations of k-copwin graphs, *Discrete Math.*, **312** (2012) 1421-1425
2. S. Neufeld and R. Nowakowski, A game of cops and robbers played on products of graphs. *Discrete Math.*, **186** (1998) 253-268.
3. R. Tošic, On cops and robber game, *Studia Sci. Math. Hungar*, **23** (1988), 225-229.
4. M. Maamoun and H. Meyniel, On a game of policemen and robber, *Discr. Appl. Math.* **17** (1987) 307-309.
5. B. Sullivan, N. Townsend, M. Werzanski, The 3x3 rooks graph ($K_3 \square K_3$) is the unique smallest graph with lazy cop number 3, *arXiv preprint, arXiv: 1606.08485*.
6. A. Bonato and R. Nowakowski, *The Game of Cops and Robbers on Graphs*, American Mathematical Society, Providence, RI, 2011.
7. M. Aigner and M. Fromme, A game of cops and robbers, *Discrete Appl. Math.* **8** (1984) 1-12.
8. B. Sullivan, N. Townsend, M. Werzanski, An introduction to lazy cops and robbers on graphs, *Coll. Math. J.* **48** (2017) 322-333.

- *PBS Infinite Series:* “The Cops and Robbers Theorem”, youtu.be/9mJEu-j1KT0
- B. Sullivan, *Talk Math with Your Friends*, May 27, 2021, youtu.be/fWhpjl44ODM