Distinguishing graphs with tropical Weierstrass weights

16 June 2026

IBS Discrete Math Seminar

Harry Richman, NCTS

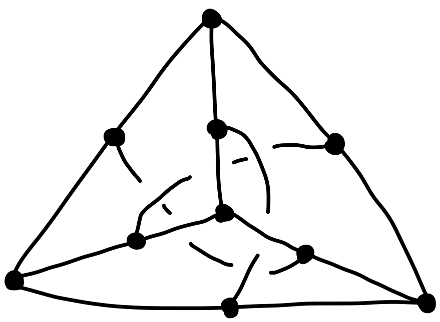
joint w/ Omid Amini & Lucas Gierczak

Paris-Saclay Marseille

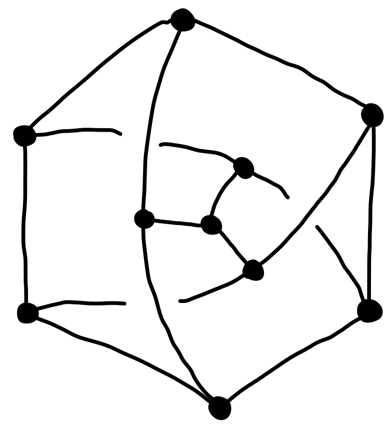
Graph Isomorphism problem

Q: Decide whether finite graphs G, H are isomorphic

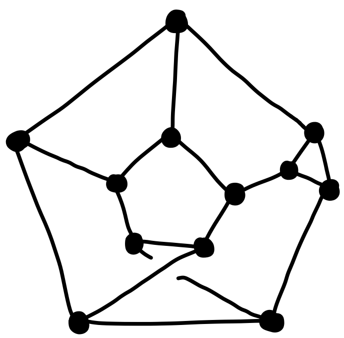
<u>Ex</u>.



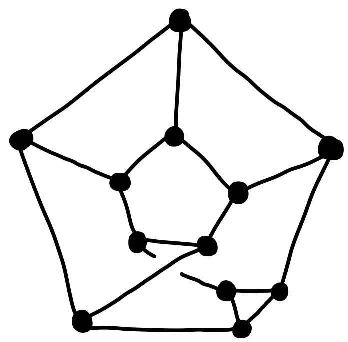
$\overset{?}{\cong}$



<u>Ex</u>.



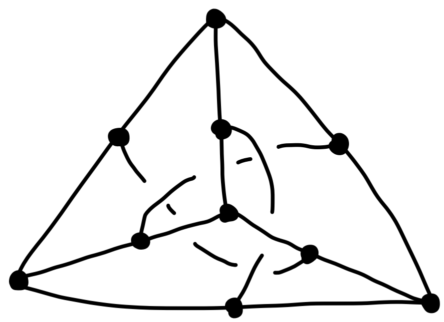
$\overset{?}{\cong}$



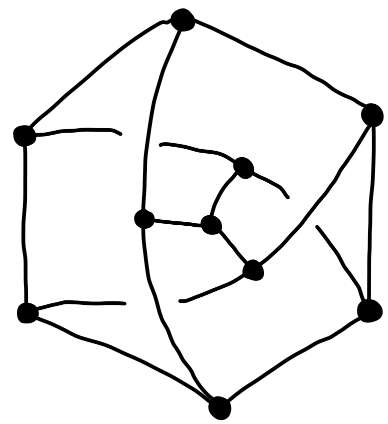
Graph Isomorphism problem

Q: Decide whether finite graphs G, H are isomorphic

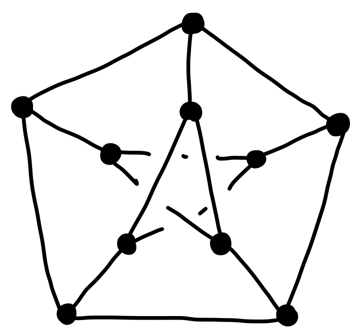
Ex.



$\cong$



$\cong$ $\cong$



Petersen graph

Graph Isomorphism problem

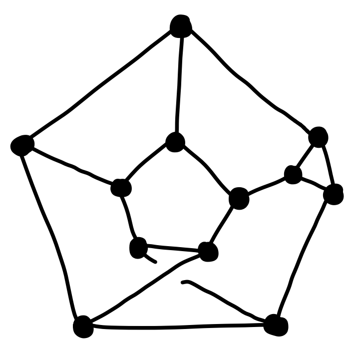
Q: Decide whether finite graphs G, H are isomorphic

$\#$ spanning trees

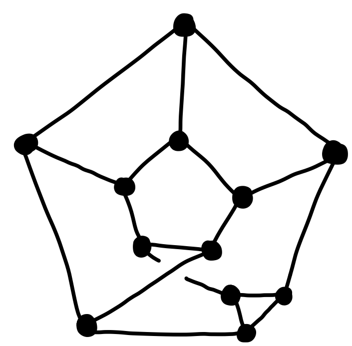
$\kappa(G) = 8100$

$\kappa(H) = 8131$

Ex.



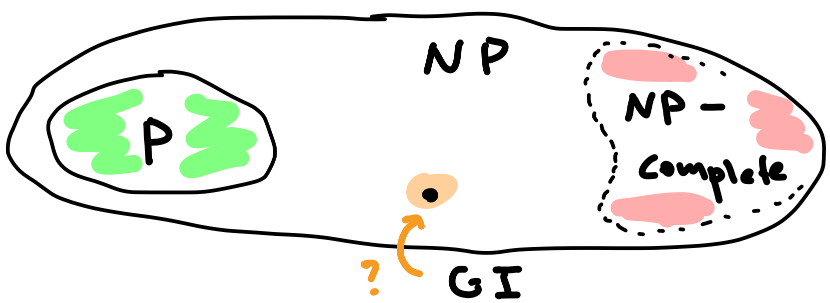
$\not\cong$



Graph Isomorphism: Complexity

- Exact complexity of Graph Isomorphism (GI) not known

- Conjecturally, GI is "NP-intermediate" i.e. neither P nor NP-complete



Rmk. Subgraph Isomorphism is known to be NP-complete.

↳ "Given (G, H), does G contain H as a subgraph?"

- (Babai 2016) Fastest known algo. for GI is $n^{O((\log n)^b)}$-time, for constant b

Graph Isomorphism: Complexity

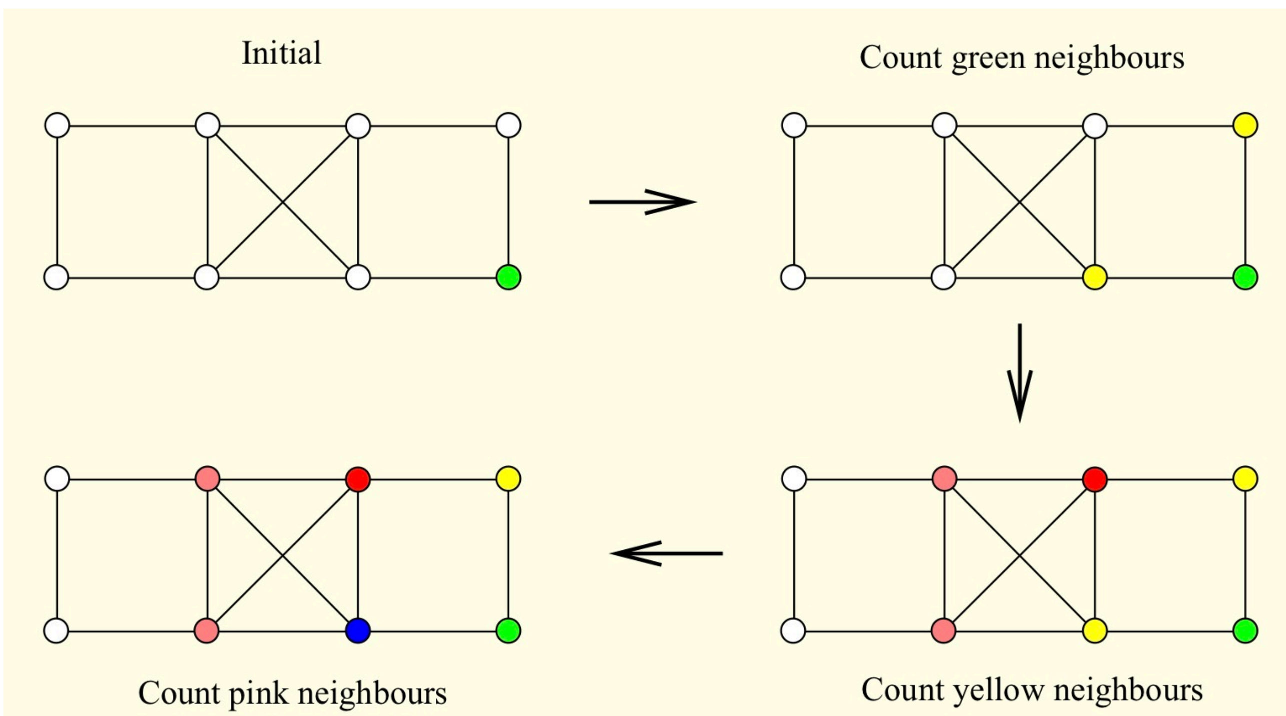
- GI known to be polynomial time on many special classes of graphs, e.g.
  - bounded degree (Luks 1982) (Grohe, Neuen, & Schweitzer 2018)
  - bounded embedding genus
    - e.g.
      - planar
      - torus-embeddable
  - excluded minors
  - excluded topological subgraphs

Rmk (McKay) "Few of these [polyn.-time] algorithms are practical, planar graphs being a notable exception"

Graph Isomorphism: Heuristics

- In practice, all current fastest implementations for GI in software use "canonical labelling" approach
  e.g. nauty (McKay), Traces (Piperno), bliss (Juntilla + Kaski), ...

- Specifically, use "individualization - refinement" paradigm on vertices



[McKay]

Graph Isomorphism: Heuristics

The Graph Isomorphism Disease*

Ronald C. Read
UNIVERSITY OF WATERLOO

Derek G. Corneil
UNIVERSITY OF TORONTO

Rmk GI used to be very popular subject in theoretical computer science (1977)

ABSTRACT

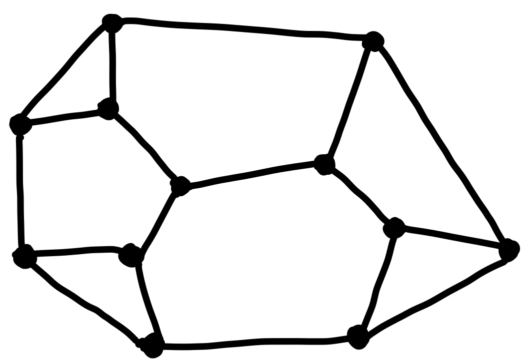
The graph isomorphism problem—to devise a good algorithm for determining if two graphs are isomorphic—is of considerable practical importance, and is also of theoretical interest due to its relationship to the concept of NP-completeness. No efficient (i.e., polynomial-bound) algorithm for graph isomorphism is known, and it has been conjectured that no such algorithm can exist. Many papers on the subject have appeared, but progress has been slight; in fact, the intractable nature of the problem and the way that many graph theorists have been led to devote much time to it, recall those aspects of the four-color conjecture which prompted Harary to rechristen it the "four-color disease." This paper surveys the present state of the art of isomorphism testing, discusses its relationship to NP-completeness, and indicates some of the difficulties inherent in this particularly elusive and challenging problem. A comprehensive bibliography of papers relating to the graph isomorphism problem is given.

Weierstrass weights

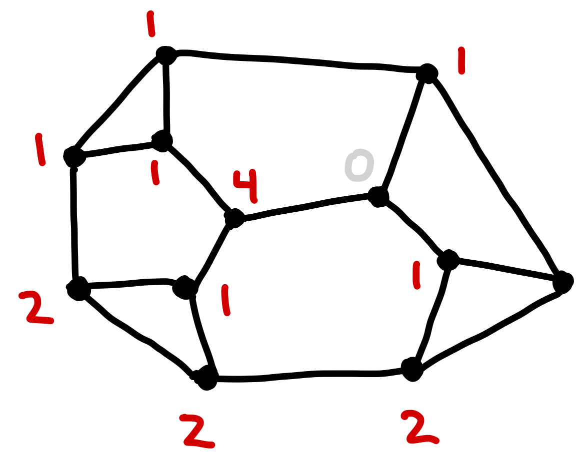
- For finite graph $G = (V, E)$, Weierstrass weights give vertex labels

$$\mu : V \to \mathbb{Z}_{\geq 0}$$

Ex.



$\rightsquigarrow$



(genus = 7)

- Labels are bounded by genus := # independent cycles (a.k.a. cyclomatic #)

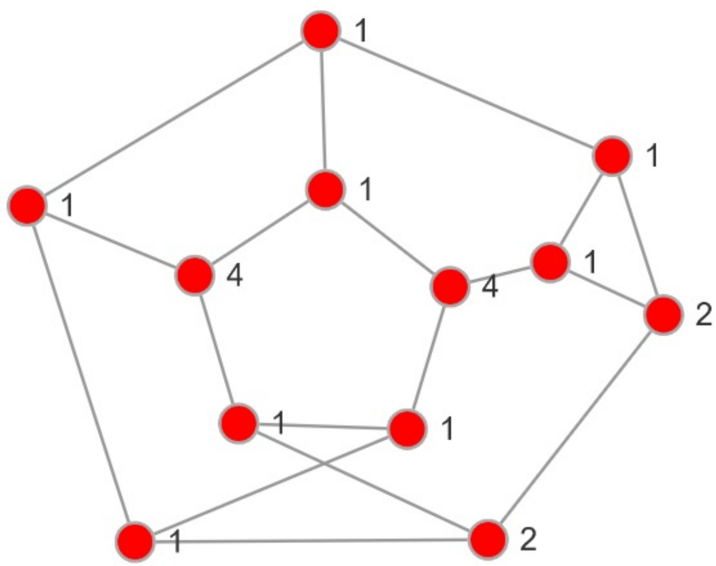
= # edges removed to get spanning tree $= |E| - |V| + 1$

Weierstrass weights

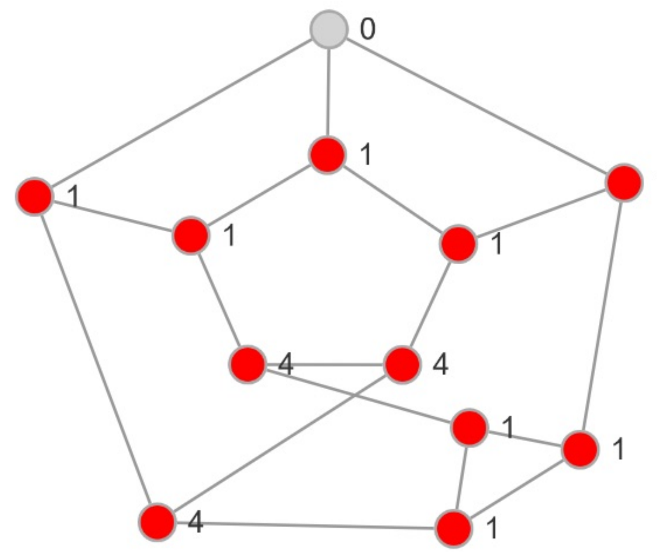
- For finite graph $G = (V, E)$, Weierstrass weights give vertex labels

$$\mu : V \to \mathbb{Z}_{\geq 0}$$

<u>Ex</u>.



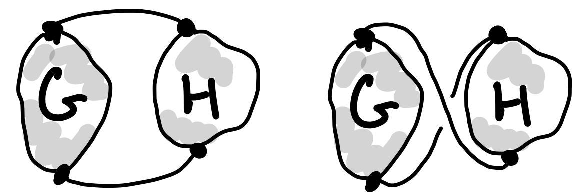
$\not\equiv$

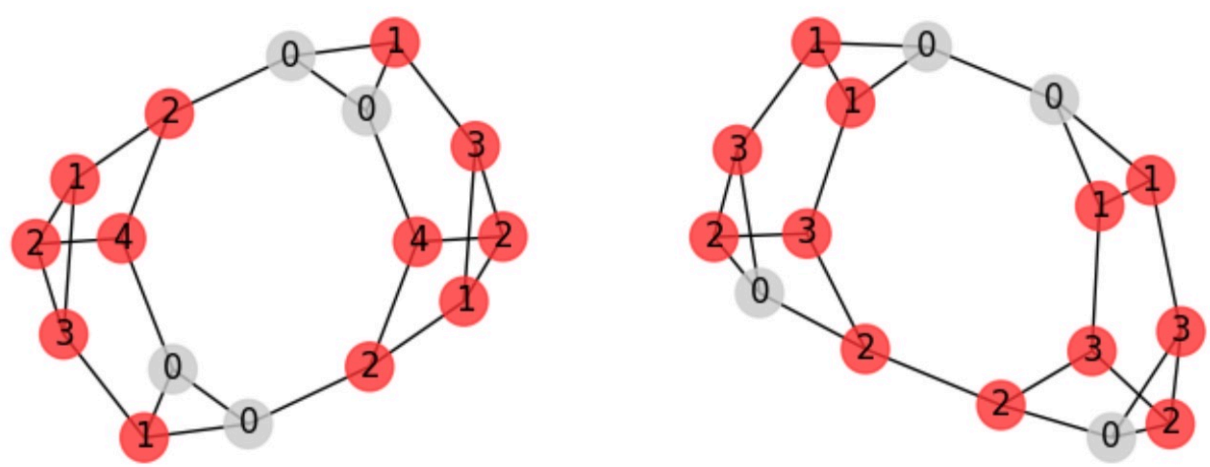


Weierstrass weights

- weights $\mu : V(G) \to \mathbb{Z}_{\geq 0}$ are isomorphism-invariant
- <u>claim</u>: (informal) weights reflect mix of local & global structure

<u>Ex</u>. "Whitney twist" of graph two-sum





Weierstrass weights : Tropical terminology

$G = (V, E)$ finite graph

- formal integer combination of $v_i \in V$ is a divisor or "chip configuration"

$$D = \sum_{v_i \in V} a_i \cdot v_i \quad \text{where} \quad a_i \in \mathbb{Z}$$

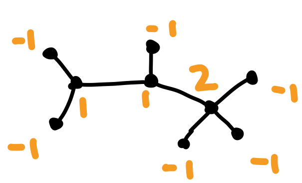
- degree of divisor $D = \sum_{v_i \in V} a_i \cdot v_i$ is $\deg D = \sum_{v_i \in V} a_i$

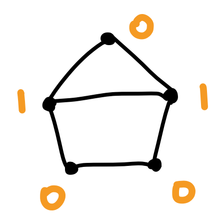
- canonical divisor of graph is

$$K = \sum_{v_i \in V} (\mathrm{val}(v_i) - 2)\, v_i \quad \text{which has} \quad \deg K = 2g - 2$$

genus

Ex.





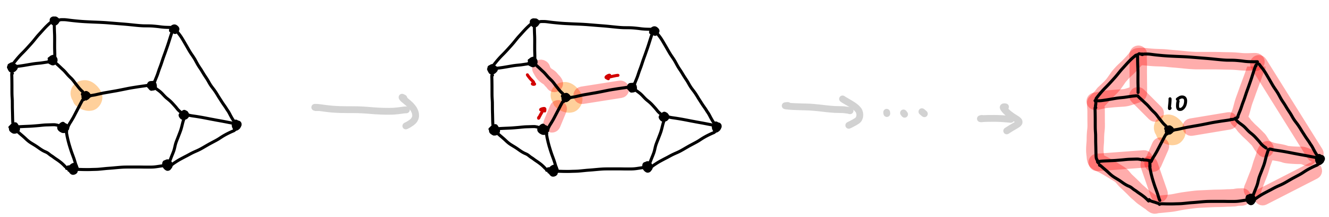
Weierstrass weights : Dhar's burning algorithm

How to compute Weierstrass weights?

- For chosen vertex v,
  - Compute **reduced divisor** at v via Dhar's algorithm
  - weight

$$\mu(v) = (v\text{-coeff.}) - (\underbrace{g}_{\text{genus}} - 1)$$

Ex.



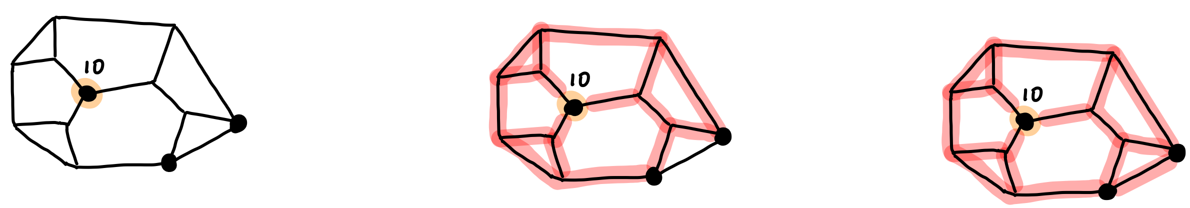
→ DONE, reduced divisor

genus = 7, so $\mu(v) = 10 - 6 = 4$

Weierstrass weights : Dhar's burning algorithm

Ex.

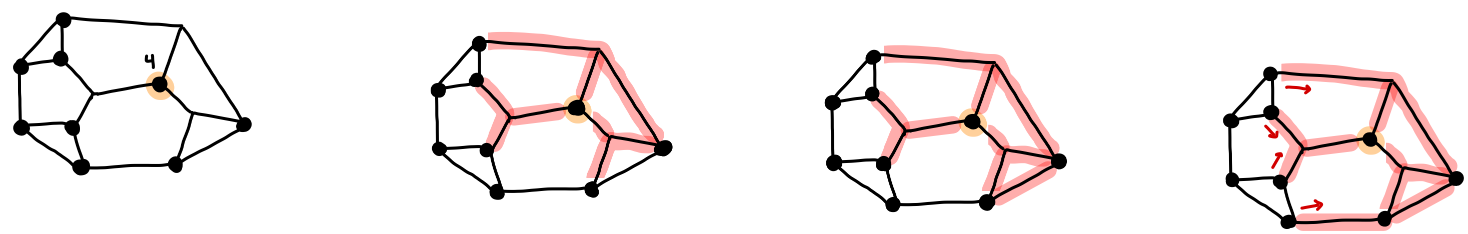
$$\mu(v) = 10 - (g-1)$$
$$= \underline{4}$$

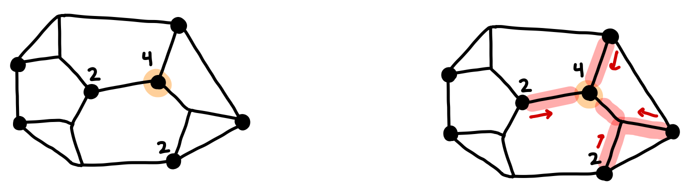


→ DONE

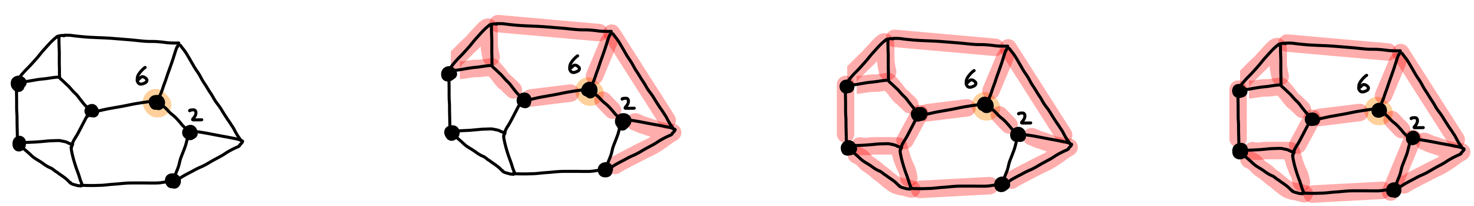
Weierstrass weights : Dhar's burning algorithm

Ex.





$$\mu(v) = 6 - (g-1)$$
$$= 0$$



→ DONE

Weierstrass weights : Running time

- To compute Weierstrass weight at one vertex :

  1. Run "fire spreading" by BFS $O(|V| + |E|)$
  2. If fire contained on proper subgraph :
  3. Apply "chip-firing" on blocking vertices
  4. Return to 1.

  $\times\ O(\text{diameter})$

  5. Else, fire spreads to whole G
  6. Output chip configuration

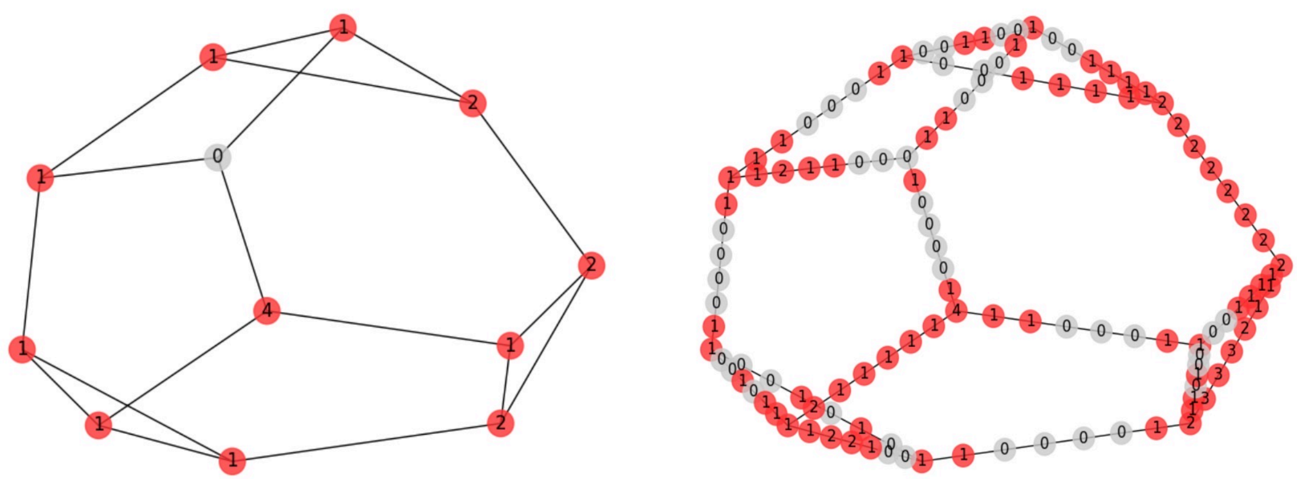
- Worst-case time $O(|V|^2 + |V| \cdot |E|)$
- Average-case for random graphs $O(|V| \log |V| + |E| \log |V|)$
- To compute weights at all vertices : don't start over

Weierstrass weights : Bounds & constraints

Prop For any vertex, Weierstrass weight $\mu(v) \le g-1$

↳ genus

Fact. Weierstrass weights are unchanged by "uniform refinement" of edges

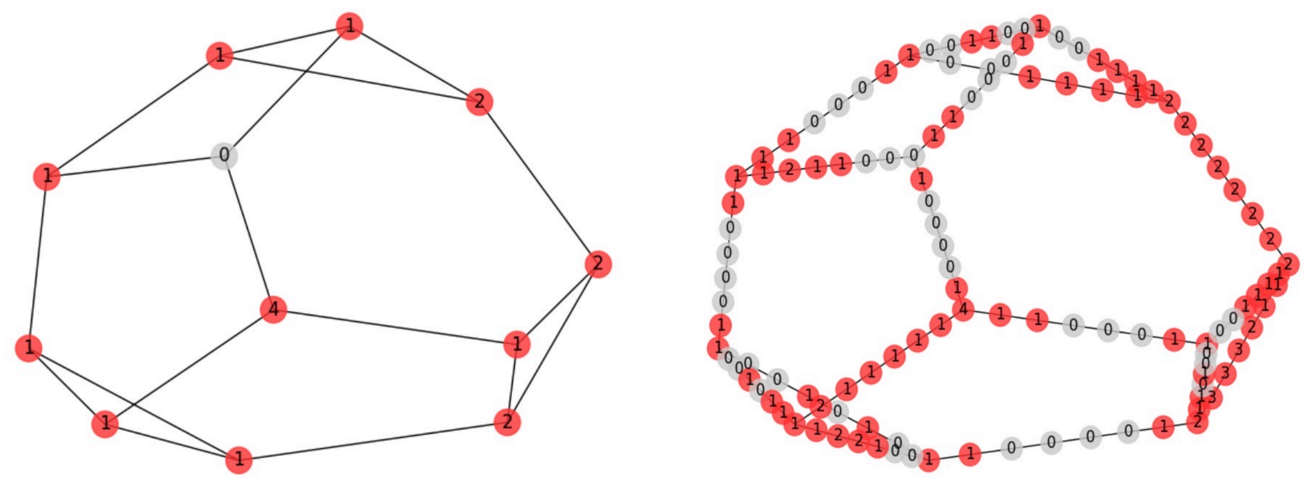


Weierstrass weights : Bounds & constraints

Prop For any vertex, Weierstrass weight $\mu(v) \leq g-1$

↳ genus

Fact. Weierstrass weights are unchanged by "uniform refinement" of edges

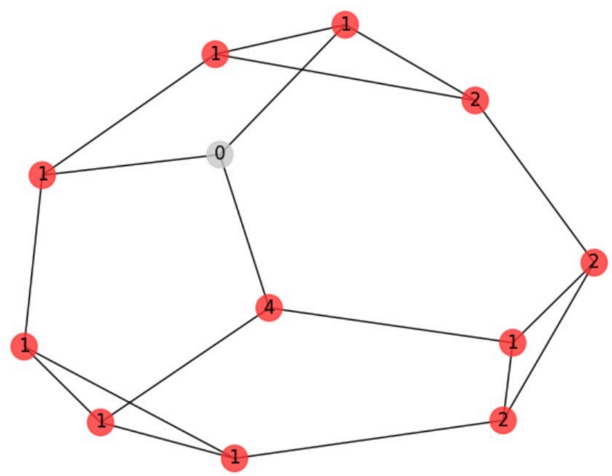


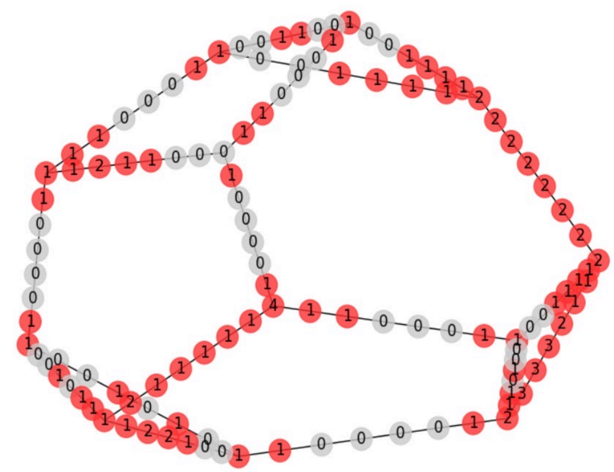
Fact Weierstrass weights are well-defined on associated metric graph, i.e. continuous space w/ edge $\simeq$ unit interval

Weierstrass weights : Bounds & constraints

Prop For any vertex, Weierstrass weight $\mu(v) \le g-1$

↳ genus





On underlying metric graph.

Thm (Amini - Gierczak - R 2026)

a) Finitely many connected comp. w/ "Weierstrass-condition" $\le g^2-1$

b) "Well-defined" weight on connected components, s.t.

$$\sum_i \mu(A_i) = g^2-1 .$$

Weierstrass weights : Bounds & constraints

On underlying metric graph.

Thm (Amini - Gierczak - R 2026)

a) Finitely many connected comp. w/ "Weierstrass-condition"

b) "well-defined" weight on connected components, s.t.

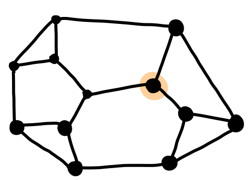
$$\sum_i \mu(A_i) = g^2 - 1 .$$

· Weight formula :

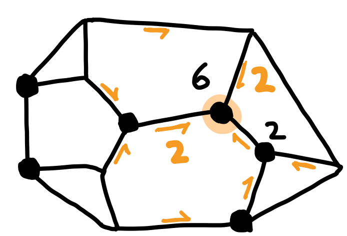
$$\mu(A) = (g+1)(g(A)-1) - \sum_{v \in \partial A} (s_v^{\nu}(K) - 1)$$

"slopes" of red.-div. of K

Ex.



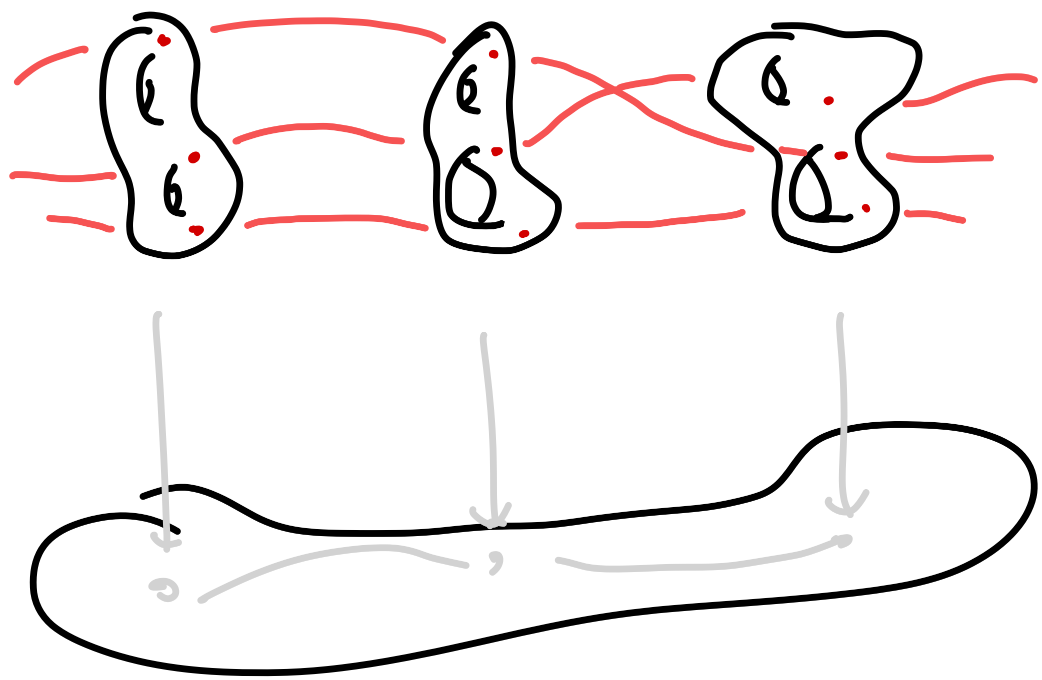
⇝ Reduced divisor



Why Weierstrass points?

Goal: understand moduli space of algebraic curves $\mathcal{M}_g$

$$\begin{array}{c} \mathcal{C}_g \\ \downarrow \\ \mathcal{M}_g \end{array}$$

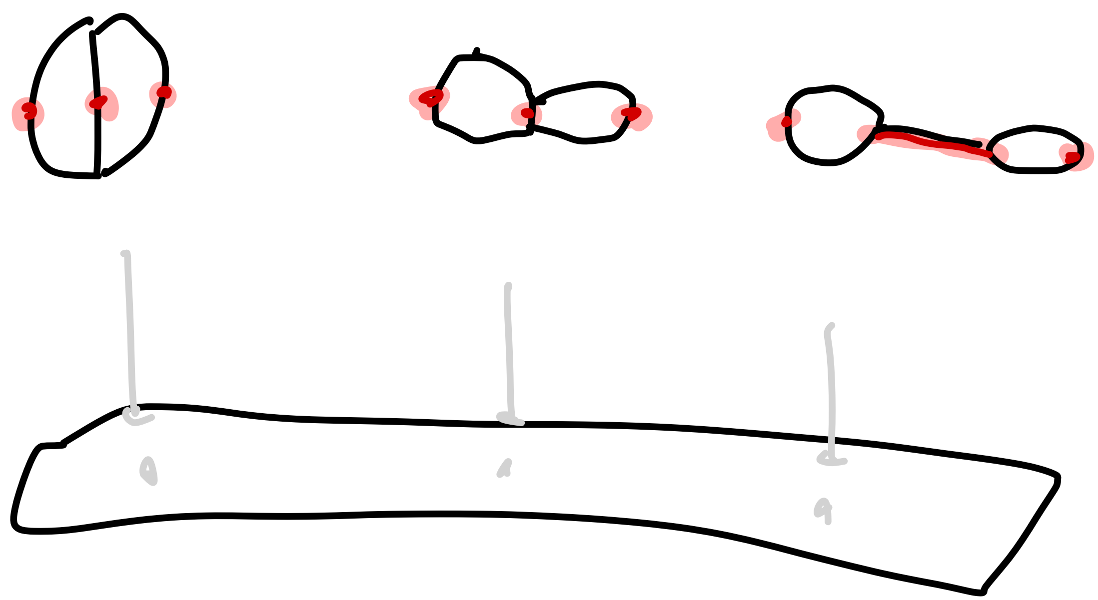


Slogan: "Weierstrass pts are coordinates on $\mathcal{M}_g$"

Tropical Weierstrass weights

Problem. What is "good" notion of tropical weight?

$$\begin{array}{c} \mathcal{G}_g^{trop} \\ \downarrow \\ \mathcal{M}_g^{trop} \end{array}$$



Fact. Tropical Weierstrass locus can be infinite, containing positive-length segments

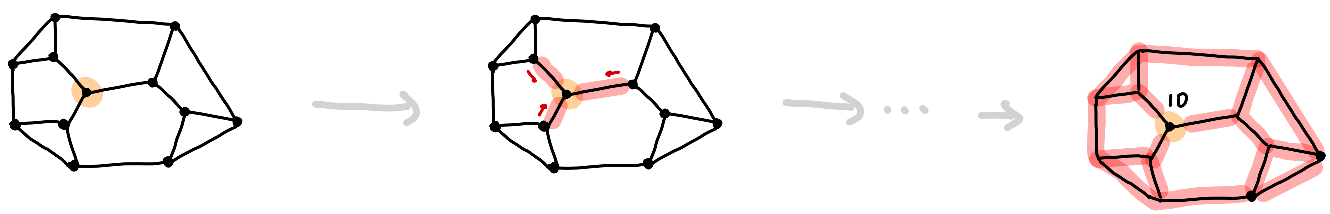
Graph isomorphism

Recap:

- Can Weierstrass weights improve GI running time?
  - Probably no in worst-case, arbitrary graphs
  - Maybe yes for certain families, or as heuristic
  - For highly symmetric / vertex transitive graphs, consider W-weights of edge-deletions

- Can we use "Jacobian" group structure on graph vertices, in other ways?

Distinguishing graphs with tropical Weierstrass weights



Thanks for listening!