

Knowledge management in the House of Graphs

Gauvain Devillez¹, Sven D'Hont², Jan Goedgebeur²

Computers in Scientific Discover 11, Kranjska Gora, Slovenia

¹University of Mons



²KU Leuven

The logo for KU Leuven consists of a dark blue rectangular box with a lighter blue border. Inside the box, the words "KU LEUVEN" are written in white, bold, uppercase, sans-serif font.

What is it ?



The House of Graphs



Database of Interesting Graphs

Most graph theorists will agree that among the vast number of graphs that exist, there are only a few thousand that can be considered really *interesting*.

It is the aim of this House of Graphs project to find a workable definition of 'interesting' and provide a [searchable database](#) of graphs that conform to this definition. Note that the searchable database does *not* contain all graphs of a given order, but only the ones which are considered 'interesting' by at least one of the users. We also allow users to add additional graphs which they find interesting. In order to avoid abuse, only [registered](#) users can add new graphs.

Lists of Graphs and Generators

We would also like to serve as a central repository for complete lists of graphs (which can be downloaded in several formats) of various graphs classes and the generators which can be used to generate these graphs. These lists of graphs and generators can be found in the [graph meta-directory](#).



Search Graphs

Browse and filter the database of interesting graphs



Draw a Graph

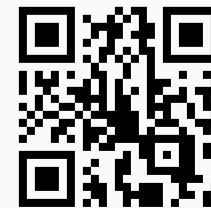
Create and visualize graphs with our interactive drawing tool



Meta-directory

Access complete lists of graphs and generators by graph class

- Online database.
- More than 600 registered users.
- Centralizes knowledge in graph theory.
- First created in 2010 [1].
- Entirely rewritten in 2022 [2].



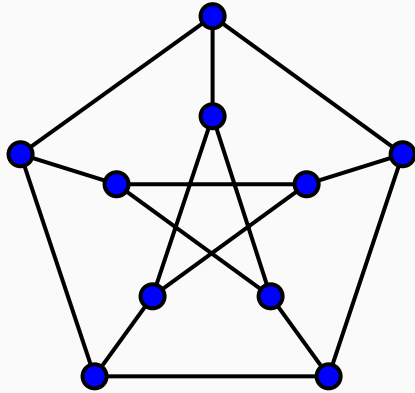
<https://houseofgraphs.org>

Vertices	Girth ≥ 3	Girth ≥ 4	Girth ≥ 5	Girth ≥ 6	Girth ≥ 7	Girth ≥ 8	Girth ≥ 9
4	1	0	0	0	0	0	0
6	2	1	0	0	0	0	0
8	5	2	0	0	0	0	0
10	19	6	1	0	0	0	0
12	85	22	2	0	0	0	0
14	509	110	9	1	0	0	0
16	4060	792	49	1	0	0	0
18	41301	7805	455	5	0	0	0
20	510489	97546	5783	32	0	0	0
22	7319447	1435720	90938	385	0	0	0
24	117940535	23780814	1620479	7574	1	0	0
26	2094480864	432757568	31478584	181227	3	0	0
28	40497138011	8542471494	656783890	4624501	21	0	0
30	845480228069	181492137812	14621871204	122090544	546	1	0
32	18941522184590	4127077143862	345975648562	3328929954	30368	0	0

- A centralized url to host or redirect lists of graphs
- Some *small* collections are hosted.
- Links to authors and their websites

Figure 2: Collections of cubic graphs

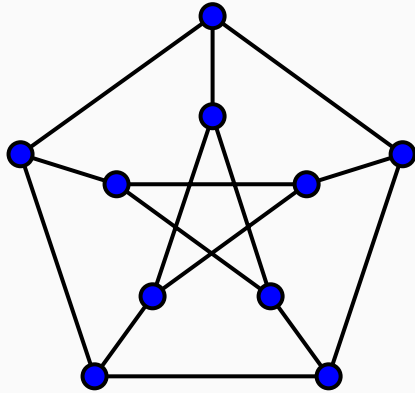
The core idea of the House of Graphs is to store **interesting graphs**.



- smallest possible snark

Figure 3: The Petersen Graph

The core idea of the House of Graphs is to store **interesting graphs**.



- smallest possible snark
- unique $(3, 5)$ -cage

Figure 3: The Petersen Graph

The core idea of the House of Graphs is to store **interesting graphs**.

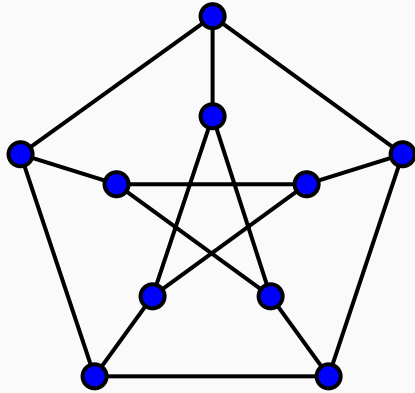


Figure 3: The Petersen Graph

- smallest possible snark
- unique $(3, 5)$ -cage
- the 10-vertices C_4 -free graph having the most number of C_5 s

The core idea of the House of Graphs is to store **interesting graphs**.

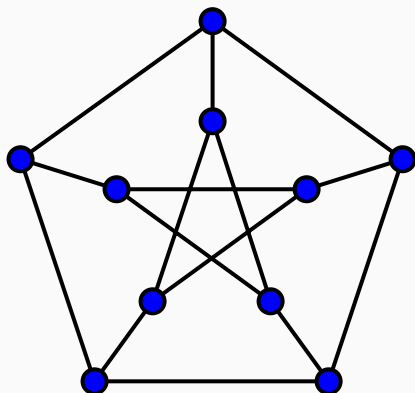


Figure 3: The Petersen Graph

- smallest possible snark
- unique $(3, 5)$ -cage
- the 10-vertices C_4 -free graph having the most number of C_5 s
- smallest graph of cop number 3

The core idea of the House of Graphs is to store **interesting graphs**.

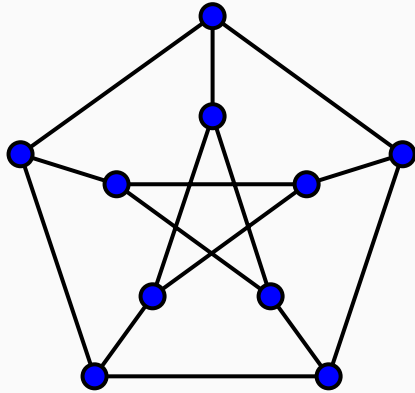


Figure 3: The Petersen Graph

- smallest possible snark
- unique $(3, 5)$ -cage
- the 10-vertices C_4 -free graph having the most number of C_5 s
- smallest graph of cop number 3
- largest cubic graph with diameter 2

The core idea of the House of Graphs is to store **interesting graphs**.

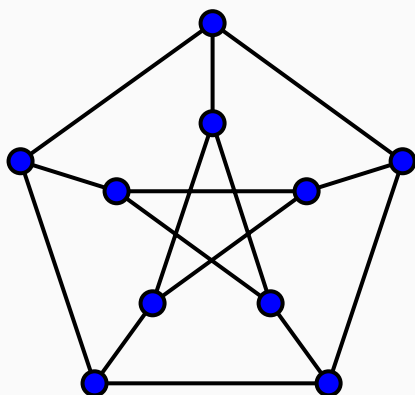


Figure 3: The Petersen Graph

- smallest possible snark
- unique $(3, 5)$ -cage
- the 10-vertices C_4 -free graph having the most number of C_5 s
- smallest graph of cop number 3
- largest cubic graph with diameter 2

Definition (informal)

A graph is said to be *interesting* if

- it appears in the **literature**,
- it contained in some **lists** (e.g. websites),
- it was pointed out by a **conjecture-making system**,
- or it is being used by **you** or other researchers.

The core idea of the House of Graphs is to store **interesting graphs**.

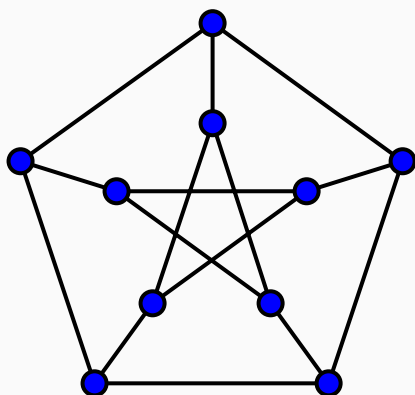


Figure 3: The Petersen Graph

- smallest possible snark
- unique $(3, 5)$ -cage
- the 10-vertices C_4 -free graph having the most number of C_5 s
- smallest graph of cop number 3
- largest cubic graph with diameter 2

Definition (informal)

A graph is said to be *interesting* if

- it appears in the **literature**,
- it contained in some **lists** (e.g. websites),
- it was pointed out by a **conjecture-making system**,
- or it is being used by **you** or other researchers.

More after the break!

- CSD5 (Sheffield, 2010, Mélot): First introduction of the House of Graphs.

- CSD5 (Sheffield, 2010, Mélot): First introduction of the House of Graphs.
- CSD10 (Kortrijk, 2024): Presentation of the new House of Graphs.

- CSD5 (Sheffield, 2010, Mélot): First introduction of the House of Graphs.
- CSD10 (Kortrijk, 2024): Presentation of the new House of Graphs.
- CSD11 (Kranjska Gora, 2026): Dive below the surface.

- CSD5 (Sheffield, 2010, Mélot): First introduction of the House of Graphs.
- CSD10 (Kortrijk, 2024): Presentation of the new House of Graphs.
- CSD11 (Kranjska Gora, 2026): Dive below the surface.

Rather than showing how to use the House of Graphs, this talk explains how it works under the hood and focuses giving an understanding on how we manage the knowledge it contains.

- What is it ? 1
- What to store? 5
- Using this knowledge 10
- The test of time 14
- Data stability 16
- Conclusion 20

What to store?

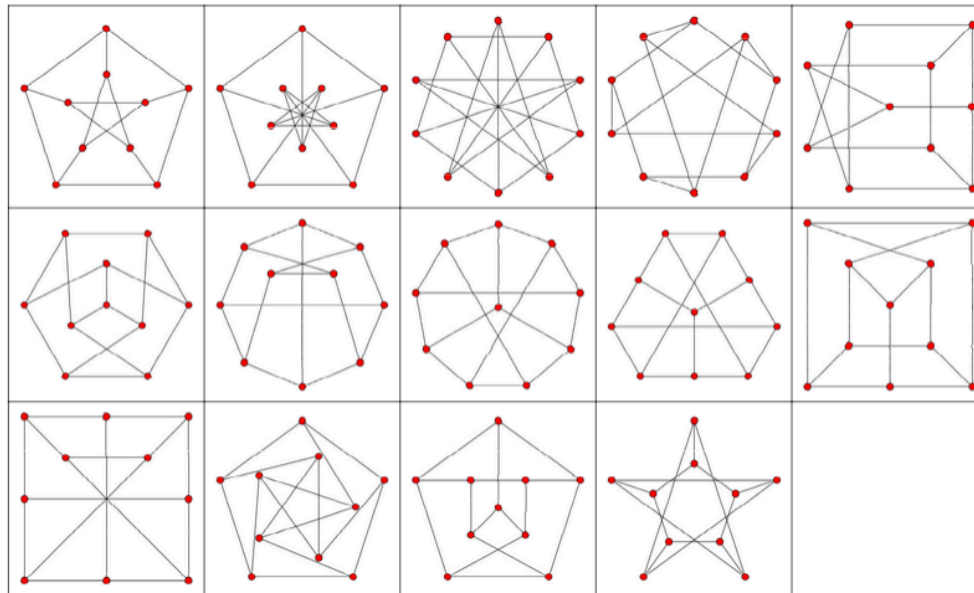
1. Gather named graphs in the literature.

Dataset construction

1. Gather named graphs in the literature.
 - All named graphs from Wolfram MathWorld (kindly provided by Eric Weisstein)

Petersen Graph

Download
Wolfram Notebook



The Petersen graph is the [cubic graph](#) on 10 vertices and 15 edges which is the unique $(3, 5)$ -[cage graph](#) (Harary 1994, p. 175), as well as the unique $(3, 5)$ -[Moore graph](#). It can be constructed as the [graph expansion](#) of $5 P_2$ with steps 1 and 2, where P_2 is a [path graph](#) (Biggs 1993, p. 119). [Excising](#) an edge of the Petersen graph gives the 4-[Möbius ladder](#) Y_3 . It is illustrated above in several embeddings (cf. Bermond *et al.* 1986; D'Angelo and Saaty and Kainen 1986; Harary 1994, p. 89; West 2000, p. 229; Knuth 2008, p. 39).



Figure 4: The Petersen graph on MathWorld

1. Gather named graphs in the literature.
2. Use GraPHedron to compute other interesting graphs (see PHOEG [3])

1. Gather named graphs in the literature.
2. Use GraPHedron to compute other interesting graphs (see PHOEG [3])

More on Thursday!

1. Gather named graphs in the literature.
2. Use GraPHedron to compute other interesting graphs (see PHOEG [3])

Total: **1500 graphs**

1. Gather named graphs in the literature.
2. Use GraPHedron to compute other interesting graphs (see PHOEG [3])

Total: **1500 graphs**

3. Open submission to the public in 2012 (CSD6, Portoroz).

1. Gather named graphs in the literature.
2. Use GraPHedron to compute other interesting graphs (see PHOEG [3])

Total: **1500 graphs**

3. Open submission to the public in 2012 (CSD6, Portoroz).

After 16 years: **28000 graphs**

User contributions

Graph name

If the graph has a specific name, please enter it below. This field is optional.

0/255

Graph info*

Please give some more information about this graph. Where did you find this graph? What does this graph represent? What makes this graph special? By adding keywords, other users can easily find this graph.

🔗 Markdown supported ▾

0/1024

Interesting invariants

Please indicate for which invariants this graph is interesting according to you. The definitions of the invariants can be found [here](#).

- | | | | |
|---|---|--|--|
| <input type="checkbox"/> Acyclic | <input type="checkbox"/> Algebraic Connectivity | <input type="checkbox"/> Average Degree | <input type="checkbox"/> Bipartite |
| <input type="checkbox"/> Chromatic Index | <input type="checkbox"/> Chromatic Number | <input type="checkbox"/> Circumference | <input type="checkbox"/> Claw-Free |
| <input type="checkbox"/> Clique Number | <input type="checkbox"/> Connected | <input type="checkbox"/> Degeneracy | <input type="checkbox"/> Density |
| <input type="checkbox"/> Diameter | <input type="checkbox"/> Domination Number | <input type="checkbox"/> Edge Connectivity | <input type="checkbox"/> Eulerian |
| <input type="checkbox"/> Feedback Vertex Set Number | <input type="checkbox"/> Genus | <input type="checkbox"/> Girth | <input type="checkbox"/> Group Size |
| <input type="checkbox"/> Hamiltonian | <input type="checkbox"/> Hypohamiltonian | <input type="checkbox"/> Hypotractable | <input type="checkbox"/> Independence Number |

Figure 4: Submitting a graph

In addition to invariant values, the following meta-data is available:

- Additional comments about why this graph is interesting
- One or several drawings:
 - Users can provide their own drawings
 - One or two (if planar) can be automatically generated
- Link to the graph's complement or line graph if present in the House of Graphs

In addition to invariant values, the following meta-data is available:

- Additional comments about why this graph is interesting
- One or several drawings:
 - Users can provide their own drawings
 - One or two (if planar) can be automatically generated
- Link to the graph's complement or line graph if present in the House of Graphs

Those are different degrees of user interaction!

In addition to invariant values, the following meta-data is available:

- Additional comments about why this graph is interesting
- One or several drawings:
 - Users can provide their own drawings
 - One or two (if planar) can be automatically generated
- Link to the graph's complement or line graph if present in the House of Graphs

Those are different degrees of user interaction!

Advice

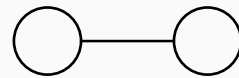
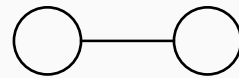
We need to decide how much control we need over the data.

Invariants are computed on the server because:

- Not all invariants have the same interest.
- They must be correct.
- They must be coherent with other graphs.

Invariants are computed on the server because:

- Not all invariants have the same interest.
- They must be correct.
- They must be coherent with other graphs.

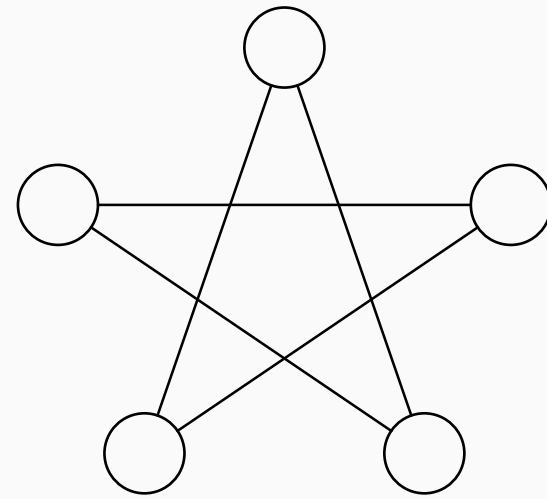
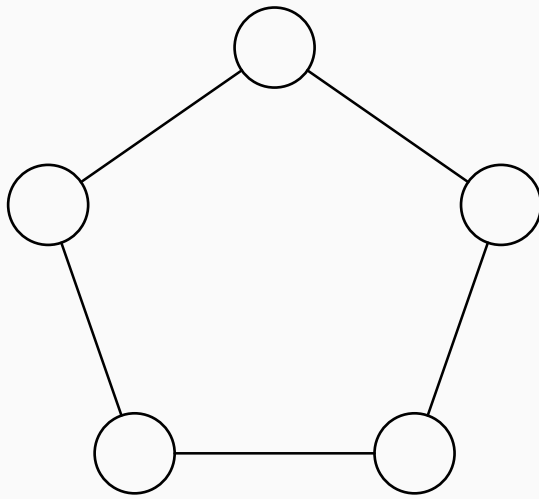


Diameter: undefined

Average Distance: ∞

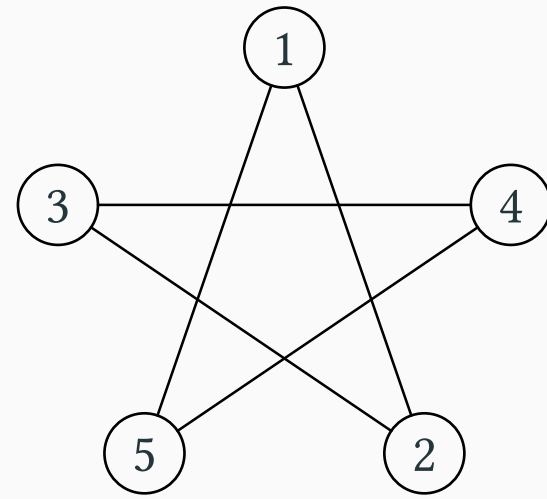
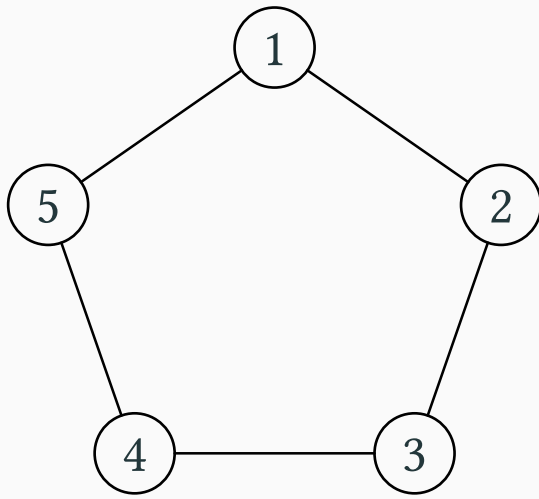
Canonical labelling

To know if two graphs are isomorphic, one method is the **canonical labelling**.



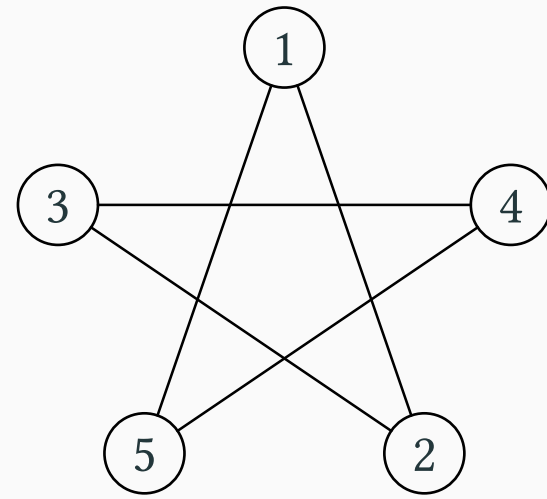
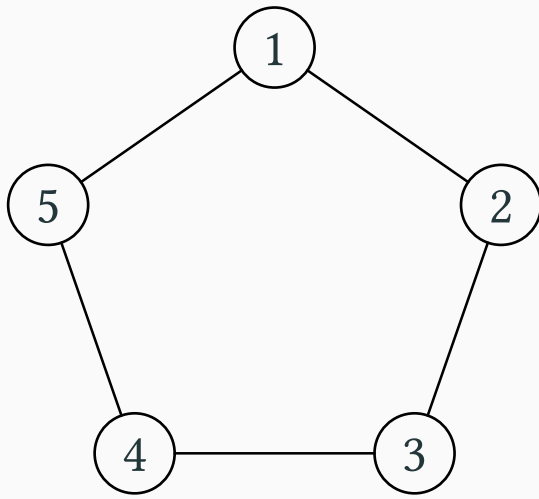
Canonical labelling

To know if two graphs are isomorphic, one method is the **canonical labelling**.



Canonical labelling

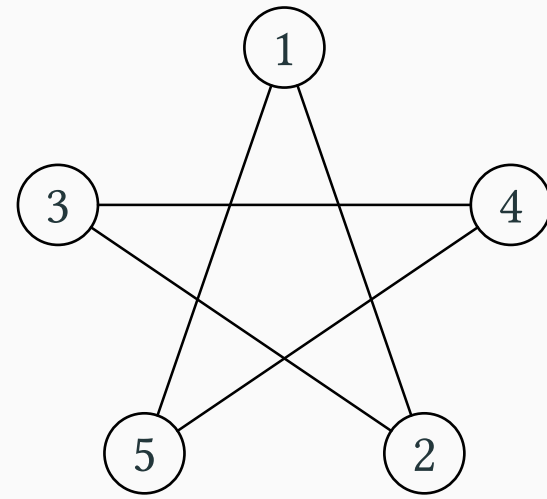
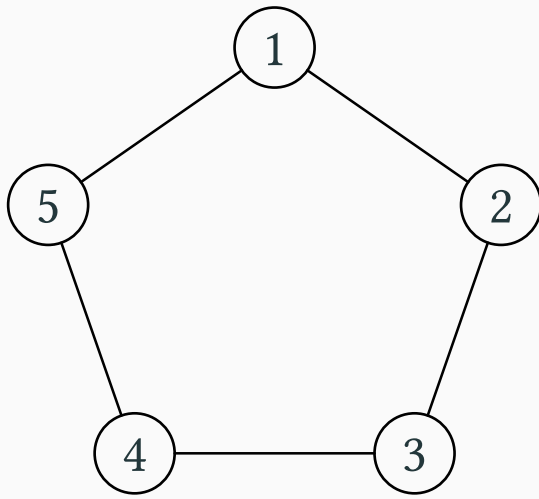
To know if two graphs are isomorphic, one method is the **canonical labelling**.



Graph6: **DqK**

Canonical labelling

To know if two graphs are isomorphic, one method is the **canonical labelling**.



Graph6: **DqK**

We use Nauty [4] to compute it.

Using this knowledge



Searching the database

Search Graphs

Combine Search Criteria

Combine any of the following criteria to create a powerful search query

Require a specific (numeric) value for an invariant

Select invariant... | equal to | Value

+

Require a (numeric) value of an invariant to be in a given range

Select invariant... | between | and

+

Only consider certain classes of graphs

Select class... | Yes No

+

Require a value of an integer invariant to be odd or even

Select class... | Even Odd

+

Require certain invariants to be 'of interest'

An invariant is marked as 'interesting' when the invariant has a special value (e.g. smallest/greatest possible value for a particular class of graphs)

Select invariants 'of interest'...

+

Search for graphs associated with a given text

Note: you may use the * wildcard

Search text...

+

Search for graphs that adhere to a given formula

Note: The formulas are evaluated from top to bottom

▲ ▼ Add formula

+

Set a subgraph constraint

Search for graphs which (do not) contain a given graph as (induced) subgraph

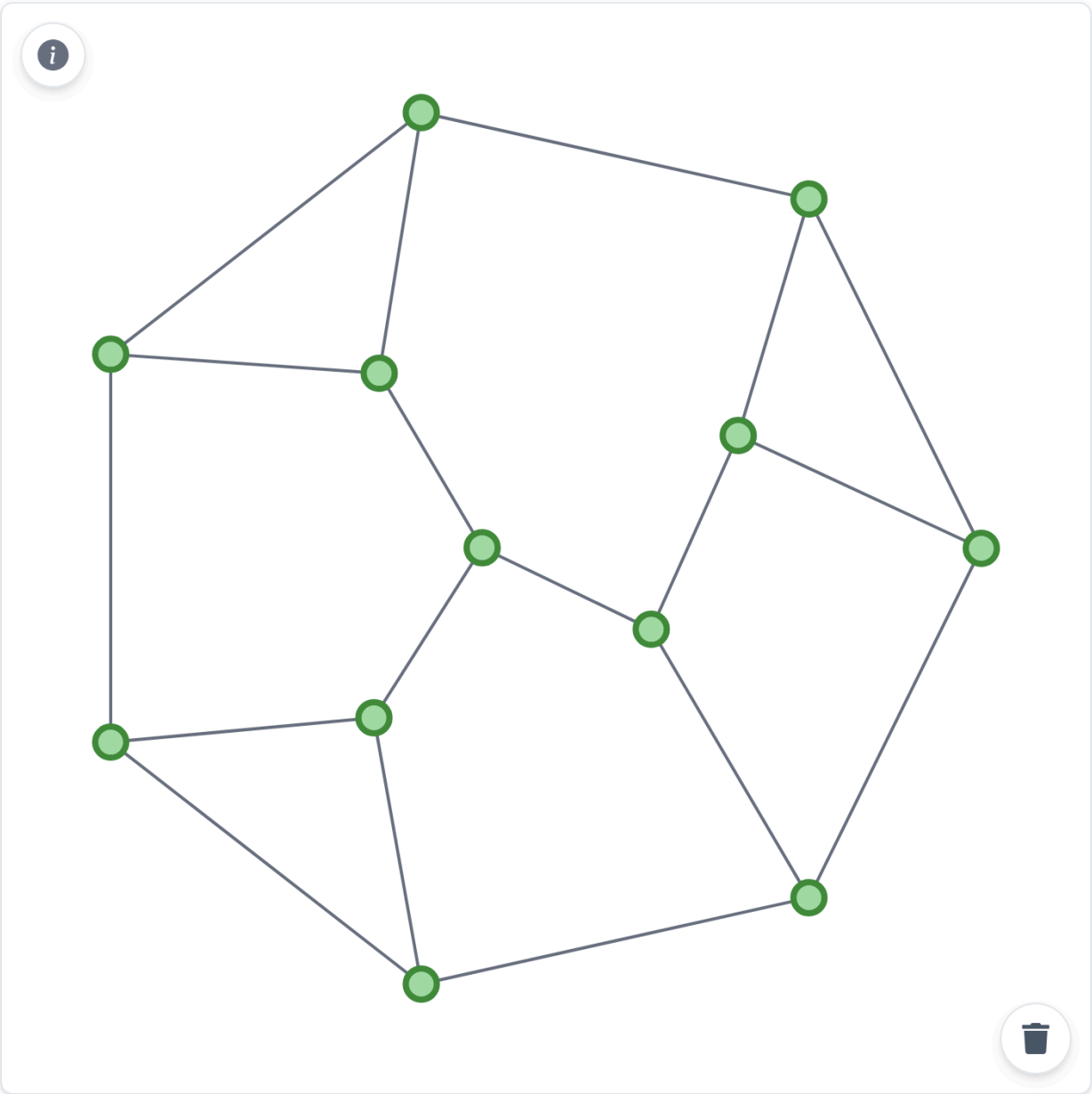
Note: Because subgraph isomorphism is an NP-complete problem, it is strongly advised to use additional constraints in the query in order to avoid a timeout.

Add graph

Search

Figure 12: Search options in the House of Graphs

Draw Graph



Delete Mode

Live

Show Labels

Pick drawing layout i

Random

Vertex size

Rotate

Mirror

x-axis y-axis

SVG

Graph6

Figure 13: Drawing a graph

Problem

Find all graphs such that maximum degree equals 4 (easy)

Problem

Find all graphs such that maximum degree equals 4 (easy)

```
SELECT graph_id from graphs  
WHERE max_deg = 4;
```

Problem

Find all graphs such that maximum degree equals 4 (easy)

```
SELECT graph_id from graphs  
WHERE max_deg = 4;
```

Problem

Find a graph from its drawing (medium)

Problem

Find all graphs such that maximum degree equals 4 (easy)

```
SELECT graph_id from graphs  
WHERE max_deg = 4;
```

Problem

Find a graph from its drawing (medium)

1. Compute its canonical labelling

Problem

Find all graphs such that maximum degree equals 4 (easy)

```
SELECT graph_id from graphs  
WHERE max_deg = 4;
```

Problem

Find a graph from its drawing (medium)

1. Compute its canonical labelling
2. Convert it to graph6

Problem

Find all graphs such that maximum degree equals 4 (easy)

```
SELECT graph_id from graphs
WHERE max_deg = 4;
```

Problem

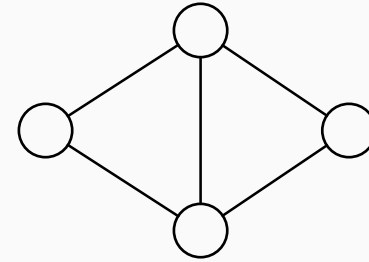
Find a graph from its drawing (medium)

1. Compute its canonical labelling
2. Convert it to graph6
3.

```
SELECT graph_id from graphs
where graph6 = '[the graph 6]';
```

Problem

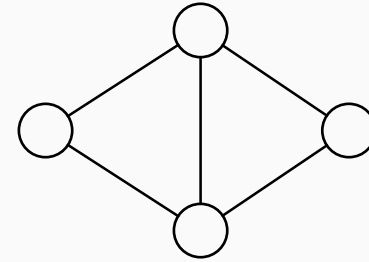
Find all graphs that contain a diamond (hard)



Problem

Find all graphs that contain a diamond (hard)

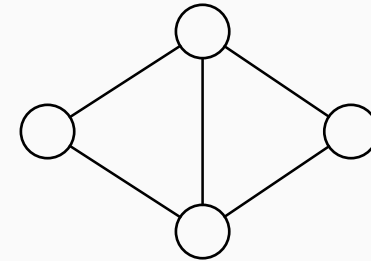
- Graphs are tested by increasing order and size
- After a given time limit, (partial-)results are returned.



Problem

Find all graphs that contain a diamond (hard)

- Graphs are tested by increasing order and size
- After a given time limit, (partial-)results are returned.
- We implemented our own version of the VF2 algorithm [5].



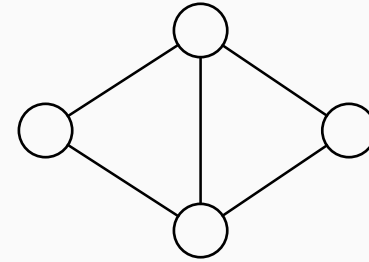
Problem

Find all graphs that contain a diamond (hard)

- Graphs are tested by increasing order and size
- After a given time limit, (partial-)results are returned.
- We implemented our own version of the VF2 algorithm [5].

Question

Why not use an existing solver?



The test of time



After development

Problem

To keep everyone happy, we have to:

- Add new functionalities
- Keep everything up to date

Problem

To keep everyone happy, we have to:

- Add new functionalities
- Keep everything up to date

Updates can go wrong. A dependency could:

- Conflict with another one.
- Be abandoned.

Problem

To keep everyone happy, we have to:

- Add new functionalities
- Keep everything up to date

Updates can go wrong. A dependency could:

- Conflict with another one.
- Be abandoned.

Advice

- Limit the number of dependencies.
- Favor software that is popular and mature.

Problem

Resources such as CPU and storage are finite.

Just because it is fast for one user, does not mean it will work with 300 simultaneous users.

Problem

Resources such as CPU and storage are finite.

Just because it is fast for one user, does not mean it will work with 300 simultaneous users.

- Graphs are limited to be between 1 and 250 vertices.
- Invariants can compute for at most 5 minutes.

Problem

Resources such as CPU and storage are finite.

Just because it is fast for one user, does not mean it will work with 300 simultaneous users.

- Graphs are limited to be between 1 and 250 vertices.
- Invariants can compute for at most 5 minutes.

Question

Why efficient invariant computation if we have a timeout?

Problem

Resources such as CPU and storage are finite.

Just because it is fast for one user, does not mean it will work with 300 simultaneous users.

- Graphs are limited to be between 1 and 250 vertices.
- Invariants can compute for at most 5 minutes.

Question

Why efficient invariant computation if we have a timeout?

- If a timeout occurs, we lack some knowledge.
- Worst offenders: genus (40% timeouts), treewidth (20% timeouts)

Data stability



Problem

Improving computation speed for invariants is important.

But:

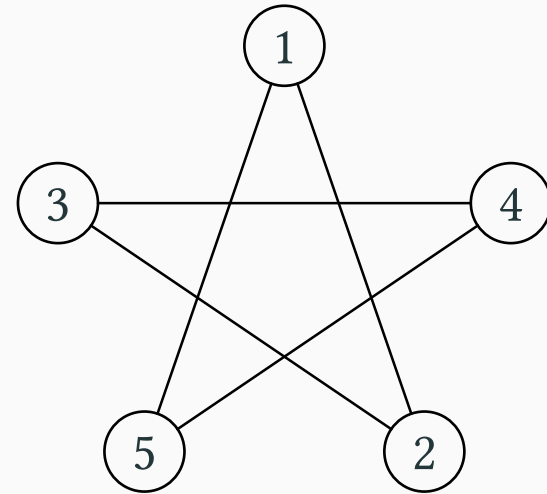
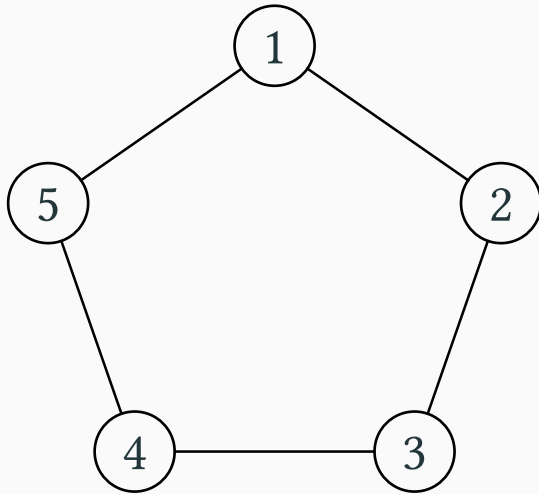
- We need a good tradeoff between speed, trust in the results and maintainability such as number of dependencies.
- The new versions must still be **coherent** with the previous ones.

Nauty updates

Another update that might speed up computation is the nauty library.

But it comes with a risk.

Nauty version 7



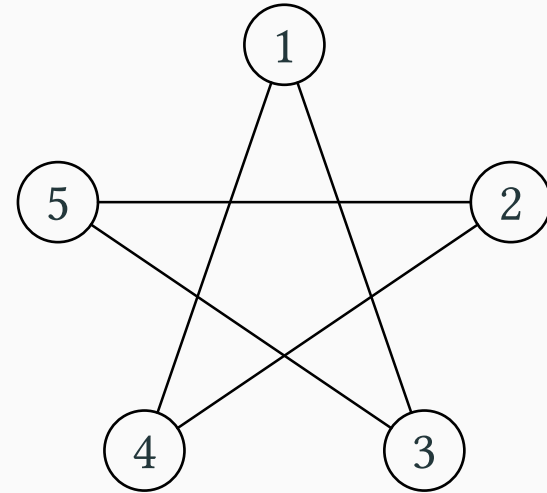
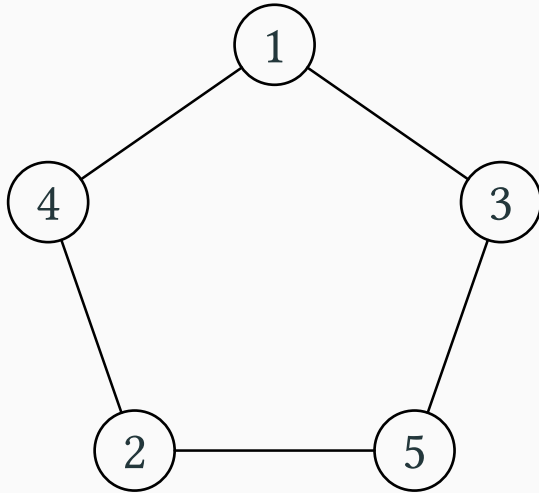
Graph6: **DqK**

Nauty updates

Another update that might speed up computation is the nauty library.

But it comes with a risk.

Nauty version 8



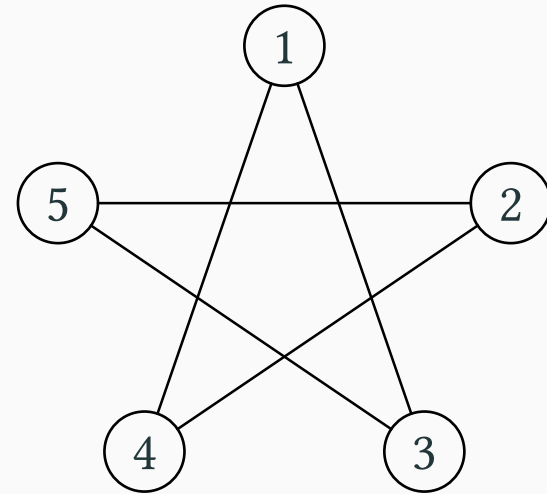
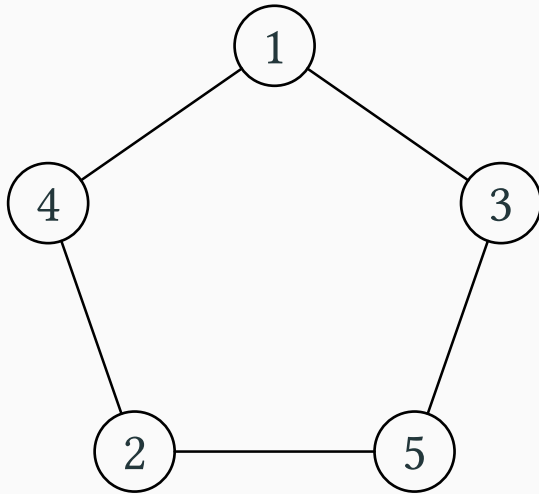
Graph6: **DUW**

Nauty updates

Another update that might speed up computation is the nauty library.

But it comes with a risk.

Nauty version 8



Graph6: **DUW**

Note: This is a possible issue but this example is hypothetical. Nauty is very stable and we never had this issue.

- A non-malicious user might cause important data to be lost.

Problem

A user deletes their account. All their graphs and comments are deleted.

- A non-malicious user might cause important data to be lost.

Problem

A user deletes their account. All their graphs and comments are deleted.

- Rather than delete them, we first shift ownerships of graphs and comments to a dummy user.
- Then we delete the account.

Problem

In a research context, if a url is published in a paper, it is not easy to edit.

Problem

In a research context, if a url is published in a paper, it is not easy to edit.

- Before 2021, the url for the Petersen graph was:

<https://hog.grinvin.org/ViewGraphInfo.action?id=660>

Problem

In a research context, if a url is published in a paper, it is not easy to edit.

- Before 2021, the url for the Petersen graph was:

<https://hog.grinvin.org/ViewGraphInfo.action?id=660>

- After the rewrite in 2022:

<https://houseofgraphs.org/graphs/660>

Problem

In a research context, if a url is published in a paper, it is not easy to edit.

- Before 2021, the url for the Petersen graph was:

`https://hog.grinvin.org/ViewGraphInfo.action?id=660`

- After the rewrite in 2022:

`https://houseofgraphs.org/graphs/660`

- But both still work today.

Advice

Keep urls stable.

Conclusion

Conclusion

- Know what to store.
- Know what users should be able to do and not to do.
- Be efficient.
- Keep dependencies low.
- Clearly identify what could impact the knowledge.
- Be stable.



<https://arxiv.org/abs/2603.23070> [6]

References

- [1] G. Brinkmann, K. Coolsaet, J. Goedgebeur, and H. Mélot, “House of Graphs: a database of interesting graphs,” *Discrete Applied Mathematics*, vol. 161, no. 1–2, pp. 311–314, 2013.
- [2] K. Coolsaet, S. D’hondt, and J. Goedgebeur, “House of Graphs 2.0: A database of interesting graphs and more,” *Discrete Applied Mathematics*, vol. 325, pp. 97–107, 2023.
- [3] S. Bonte, G. Devillez, V. Dusollier, and H. Mélot, “PHOEG: an online tool for discovery and education in extremal graph theory.” [Online]. Available: <https://arxiv.org/abs/2603.27242>
- [4] B. D. McKay and A. Piperno, “Practical graph isomorphism, II,” *Journal of symbolic computation*, vol. 60, pp. 94–112, 2014.
- [5] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, “A (sub) graph isomorphism algorithm for matching large graphs,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 10, pp. 1367–1372, 2004.
- [6] G. Devillez, S. D’hondt, and J. Goedgebeur, “Knowledge management in House of Graphs.” [Online]. Available: <https://arxiv.org/abs/2603.23070>