

©Copyright 2019

Sharat Chandra

A Complete Implementation of Graphlopedia:
A Fingerprint Database for Graphs

Sharat Chandra

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2019

Committee:

Sara Billey

Jayadev Athreya

Program Authorized to Offer Degree:
Mathematics

University of Washington

Abstract

A Complete Implementation of Graphlopedia:
A Fingerprint Database for Graphs

Sharat Chandra

Chair of the Supervisory Committee:
Sara Billey

The OEIS is a widely used and incredibly powerful tool to catalogue user-submitted sequences of interest to the mathematical community, and to associate with them a list of citations where these sequences can be found. In particular, the OEIS serves to bind together mathematical literature which may otherwise appear in very different contexts around a central, searchable feature - a numerical sequence. Billey and Tenner formalized the idea of building a database that binds literature around a searchable feature, termed a fingerprint, into the concept of a Fingerprint Database. In this project, we build a searchable encyclopedia web app and database using Django for graphs that appear in mathematical literature that we have named Graphlopedia. Graphlopedia, hosted at graphlopedia.org, uses graph isomorphism testing and image classification to combine user submitted entries with automatically crawled e-print sources, like the ArXiv, into a single, easily searchable location.

TABLE OF CONTENTS

	Page
Chapter 1: Introduction	1
1.1 What is a Mathematical Fingerprint Database?	1
1.2 The Origins of Graphlopedia: The WXML Team	3
Chapter 2: Background	5
2.1 The Realizability of Degree Sequences	5
2.2 The Effectiveness of Searching by Degree Sequence	6
2.3 Brute Force Computations of the Likelihood of False Positives	8
Chapter 3: Implementation Details	14
3.1 Searching Graphs	14
3.2 Submitting Entries	14
Chapter 4: Future Features	18
4.1 Searching by Edge Sequence	18
Chapter 5: Appendix	19
5.1 Graphlopedia NetworkGraph Schema	19
Bibliography	23

Chapter 1

INTRODUCTION

1.1 What is a Mathematical Fingerprint Database?

The OEIS, created by Dr. Neil Sloane, is a widely used and incredibly powerful tool to catalogue user-submitted sequences of interest to the mathematical community, and to associate with them a list of citations where these sequences can be found [11]. In particular, the OEIS serves to bind together mathematical literature which may otherwise appear in very different contexts around a central, searchable feature - a numerical sequence designated by an A number. For example, the OEIS sequence A000088 (the number of graphs on n unlabeled nodes) has in its entry thirteen textual references where this sequence would be found and 75 links to eprints and other sources where this sequence is relevant. These various citations are linked to one another via the OEIS entry A000088 and users interested in one citation for results that involve A000088 have reason to explore the others. Billey and Tenner formalized the idea of building a database that binds literature around a searchable feature, termed a fingerprint, into the concept of a Fingerprint Database[4].

Fingerprints may be non-unique and it is a generally poorly defined problem to ask for a ‘good’ fingerprint for a given class of mathematical results. As such, some choices need to be made to clarify what properties of a given result or mathematical object are the focus of the database, and whether those properties have a form that is both searchable, and can be meaningfully filtered over.

Some Fingerprint Databases currently operating include Dr. Bridget Tenner’s Database of Permutation Pattern Avoidance. The aim of this database is to provide a resource of phenomena characterized by avoiding a finite number of permutation patterns, and in this case, the permutation pattern serves as fingerprint. [13] These patterns extend beyond sequences

of numbers, as the Database explains, since barred permutations are valid searchable entries. Another Fingerprint Database, whose primary fingerprint is not necessarily a sequence, is FindStat - a database of combinatorial statistics.[12] Similar to OEIS, each result in FindStat houses a list of papers tied to it, and can be searched by assigning integer outputs to a sample collection Dyck paths. In this way, the statistics displayed by FindStat are a fingerprint for the papers in the database.

1.1.1 *What is Graphlopedia?*

Graphlopedia is a Fingerprint Database for graphs and networks, hosted at graphlopedia.org. The fundamental problem that this database aims to solve is that the same graph may go by many names across fields of study or be understood in many contexts and, unlike the numerical sequences or combinatorial statistics, there is not an easy way to recognize when two presentations of graphs in two different contexts represent the same graph due to the current difficult nature of computationally solving the graph isomorphism problem. At the moment, *we only intend to host simple graphs and any graphs discussed in this paper are assumed to be simple unless otherwise specified.*

The primary fingerprint we have chosen to represent our graphs is its *degree sequence*.

The degree sequence of a graph is the non-increasing list of degrees of nodes in the graph. The length of this list is therefore equal to the number of nodes in the graph.

While the degree sequence of a graph is certainly non-unique, a primary assumption of the database is that the families of graphs that appear in literature are sufficiently distinct with respect to node degrees. Since Graphlopedia prioritizes graphs from references, the number of graphs that appear in a given search ought to be sufficiently small, so that relevant graphs may be found quickly by other filtering methods, or even by manual observation. We explore the extent to which this assumption may hold in Chapter 2.

There have been a few attempts at collecting and organizing graphs into online databases in the past, such as the House of Graphs, which aims to form a workable definition of 'interesting' graphs and to serve as a repository for lists of graphs and graph generators.[5] While the House of Graphs does provide references to their graph collections, and the ability to search over their database, the searching feature is constrained to numerical invariants, like the average degree or chromatic index, and references are not as ubiquitous and integral to the database as we intend to do here. We hope that the combined collections of interesting graphs from sources like the House of Graphs and fingerprinted graphs from Graphlopedia provide a broad and rich pool of data for the benefit of anyone interested in the world of graph theory.

1.1.2 How do we continue growing?

While there are already a variety of graphs added to the database, one small group cannot hope to catalogue the vast expanse of graphs that occur in the mathematical and scientific literature. As such, we rely on the mathematical community at large to help us with this immense task. Users of Graphlopedia can submit new references and graphs to the database as they come across them in the wild.

We aim to make the process of submission as simple for the end user as possible in three primary ways. First, we provide human intuitive, graphical interfaces and seriously consider how web design choices affect user experience. Second, we require the use of publicly available tools and standards with minimal tweaks to increase interoperability with other projects. Finally, we automate tasks and verification wherever possible, out of sight of the end user to allow for future growth of the database and assurance of the reliability of the data.

1.2 The Origins of Graphlopedia: The WXML Team

Graphlopedia began as a project headed by Sara Billey in 2016, along with a dedicated team of students in the WXML, the Washington Experimental Mathematics Lab. The team included W. Riley Casper as TA, and undergraduates Kimberly Bautista, Aaron Bode, Dien

Dang, Nicholas Farn, Graham Kelley, Stanley Lai, Adharsh Ranganathan, Michael Trinh, Alex Tsun, Katrina Warner They experimented with the format, image processing of graphs, and using machine learning to recognize graphs. Their work lead to an initial version of Graphlopedia in pdf format. [3]

Chapter 2

BACKGROUND

2.1 The Realizability of Degree Sequences

Most sequences of nonnegative integers do not correspond to the degree sequences of any graph, so unlike the OEIS, searching a random sequence should not be expected to return a result. However, there are concrete criteria for when a sequence is realizable as the degree sequence of a graph.

Definition 2.1. A sequence $(d_1 \geq d_2 \geq \dots \geq d_n \geq 0)$ is called *graphic* if it can be realized as the degree sequence of a graph

Theorem 2.2 (Erdős–Gallai). *A finite, decreasing sequence $(d_1 \geq d_2 \geq \dots \geq d_n \geq 0)$ is graphic if and only if*

$$\sum_{i=1}^k d_i \leq k(k-1) + \sum_{j=k+1}^n \min(d_j, k) \quad \text{for } k = 1, \dots, n$$

This result is one of several of the same flavor that characterize the realizability of degree sequences, such as the Havel-Hakimi Algorithm. [7, 8]

Theorem 2.3 (Havel, Hakimi). *A finite, decreasing sequence $S = (d_1 \geq d_2 \geq \dots \geq d_n \geq 0)$ is graphic if and only if $S' = (d_2 - 1, d_3 - 1, \dots, d_{d_1-1}, d_{d_1+2}, d_n)$ is graphic, with S' sorted if necessary.*

Aigner and Triesch reformulated the Erdős-Gallai Theorem and the Havel-Hakimi Algorithm beautifully in terms of the *dominance lattice* of number partitions. [1] Every partition of m given by $p = (p_1 \geq p_2 \geq \dots \geq p_n \geq 0)$ with $m = p_1 + \dots + p_n$ has a corresponding Ferrer's Diagram, on which a partial ordering is defined.

The partition dominance lattice, $\mathcal{L}(m)$, is determined by the ordering

$$p' \prec p \iff \sum_{i=1}^k p'_i \leq \sum_{i=1}^k p_i \quad \text{for } k = 1, \dots, n$$

where p_i represent the length of the rows of the Ferrer's Diagram.

Definition 2.4. The *Durfee square*, $f(p)$, of a Ferrer's diagram p is the length of the largest square to fit in the diagram. Alternatively,

$$f(p) = \max\{i \mid p_i \geq i\}$$

Theorem 2.5 (Aigner-Triesch Reformulation). *Given a number e , the graphic sequences with exactly e edges, $\mathcal{G}(e)$ can be found as a lower order ideal in the dominance lattice $\mathcal{L}(2e)$. A sequence in $\mathcal{G}(e)$ is maximal in the ordering if and only if*

$$\sum_{i=1}^k (p_i - q_i) = -k \quad \text{for } k = 1, \dots, f(p)$$

where q_i are the length of columns in the Ferrer's diagram of p .

This reformulation provides a particularly clean criteria by which one can test whether a sequence is a graphic, simply by allowing $\sum_{i=1}^k (p_i - q_i) \leq -k$ in the above statement.

2.2 The Effectiveness of Searching by Degree Sequence

The primary assumption that Graphlopedia makes is that searching for a graph by degree sequence or partial degree sequence is an 'effective' way of filtering the database for the graph that one is interested in. Since there could potentially be many graphs with the same degree sequence, the end user may have to manually search through a few entries before finding the specific graph they have in mind. To this end, it is useful to quantify how many 'false positives' one may encounter.

2.2.1 Frequencies of Degree Sequences of Simple Graphs

The portion of the problem of effectively searching for a degree sequence we explore can be expressed as follows.

For small n , if one is picked at random, a graph on n vertices with degree sequence d , what is the probability distribution on the number of graphs $N(n)$ with the given degree sequence? Further, what is the growth rate of the maximum number of graphs on n vertices that share a degree sequence?

The first portion of this question rests on what one means by 'random'. In the case of Erdős–Rényi random graphs, there are several results known that answer this question in the limiting regime when n is large. [2, 6, 10] We will briefly outline the problem and results as provided by Chatterjee, Diaconis, and Sly.

Likelihood of Large Random Graphs with a Given Degree Sequence

We assume that the degree sequence is a sufficient statistic for the probability distribution on graphs and that the distributional form is exponential. From this, we may construct our distributional model from independent Bernoulli random variables.

Definition 2.6. Given $\beta = (\beta_1, \dots, \beta_n) \in R^n$, let the law P_β be defined as:

For each $1 \leq i \neq j \leq n$, put an edge between vertices i and j with probability

$$p_{ij} := \frac{e^{\beta_i + \beta_j}}{1 + e^{\beta_i + \beta_j}}$$

independently of all other edges.

Thus, if G is a graph with degree sequence (d_1, \dots, d_n) , the probability of observing G under P_β is

$$\frac{e^{\sum_i \beta_i d_i}}{\prod_{i < j} (1 + e^{\beta_i + \beta_j})}$$

Given a sample degree sequence, $d = (d_1, \dots, d_n)$, Chatterjee, Diaconis, and Sly provide a simple to compute function $\varphi : R^n \rightarrow R^n$ with

$$\varphi_i(\mathbf{x}) := \log d_i - \log \sum_{i \neq j} \frac{1}{e^{-x_j} + e^{x_i}}$$

whose fixed points corresponds to the maximum likelihood estimation for β , given by $\hat{\beta}$, with fast convergence. Further, $\hat{\beta}$ approximates β well as the number of vertices grow large.

This iterative approach allows us to quickly compute the probability of observing a graph with given degree sequence, provided that the method of generation of these graphs properly follows the law P_β . While this is indeed useful in principle for deciding how likely one may be to find a false positive in Graphlopedia, it does not provide enough detail about the likelihood of selecting a graph with a given degree sequence when the number of vertices is small (perhaps under 100 vertices) - as we would expect to be common in the database.

Thus, for the case of small graphs, we turn to some brute force methods.

2.3 Brute Force Computations of the Likelihood of False Positives

2.3.1 Likelihood of False Positives Amongst Small Graphs by Brute Force

In a scenario when one has only the degree sequence of a desired graph as a search parameter, a user may have to manually look through all graphs with that degree sequence to find what they need. If a simple graph on n vertices is chosen uniformly at random, we can model the probability distribution of the number of graphs $N(n)$ with that particular degree sequence.

We used Sage to compute the degree sequences of all simple graphs up to 9 vertices with none isolated and constructed histograms to visualize the probability distribution on $N(n)$. We have observed that the distribution appears to be Weibull with two parameters k, λ that depend on the mean and median of the data - which perhaps serves as good interpolation of the exponential distribution forms in the higher vertex regime to the small graph case.

Our limit of 9 vertices was primarily due to computation time, as the number of isomorphism classes on n unlabelled vertices grows as $2^{\binom{n}{2}}/n!$ (OEIS A000088) Here, we present $n = 8$ as an example.

Distribution of $N(8)$

```
n = 8
pn, norm_pn, avg, med = Nd_Degree_Sequence(n,d)
data = Nd_Degree_Sequence_Stats(n,d,pn, norm_pn,theta=1, xmax=200, ymax=max(norm_pn))
k, L= data[0], data[1].n()
print('k = %f, lambda = %f'%(k, L))
data[-1]
```

k = 0.507892, lambda = 8.231179

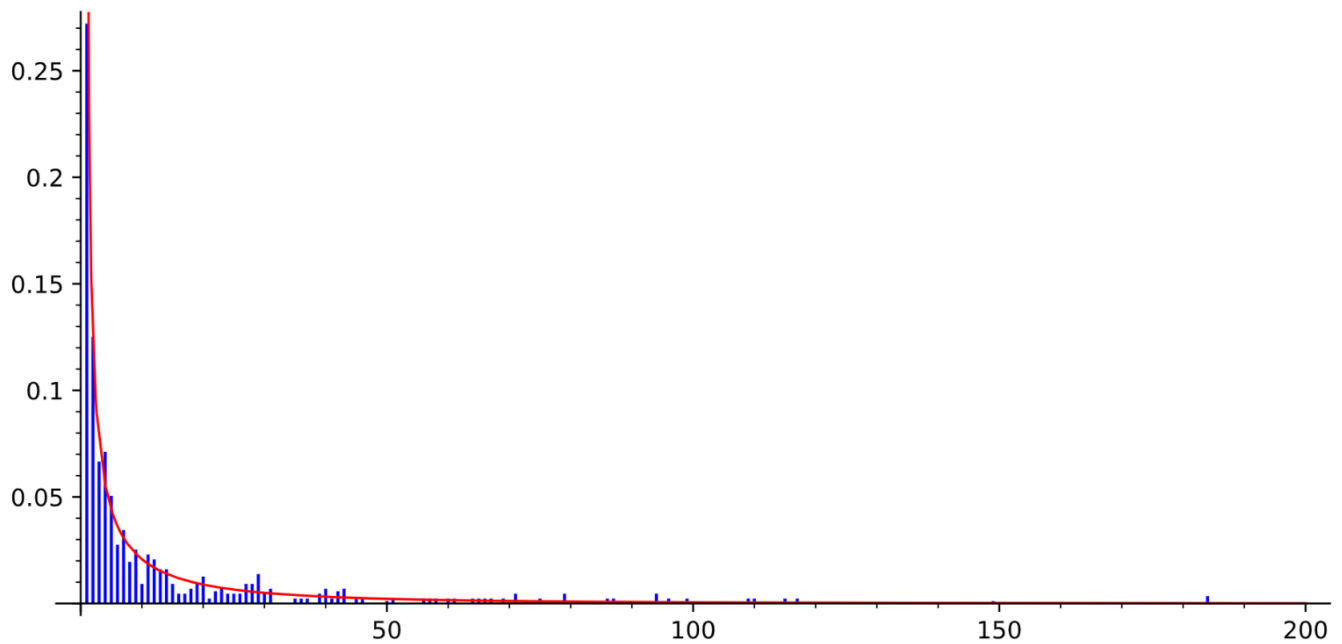


Figure 2.1: Distribution of $N(8)$

Likelihood of False Positives for $n = 8$

- Sequences with fewer than 19 graphs represent 80% of the random selections
- Sequences with fewer than 37 graphs represent 90% of the random selections
- Sequences with fewer than 65 graphs represent 95% of the random selections

- Sequences with fewer than 111 graphs represent 99% of the random selections

Unfortunately, with the limited amount of data we were able to collect, we did not find a clean relationship between the the two Weibull parameters k, λ and the number of vertices n .

2.3.2 Frequencies of Degree Sequences in Small Random Graphs by Brute Force

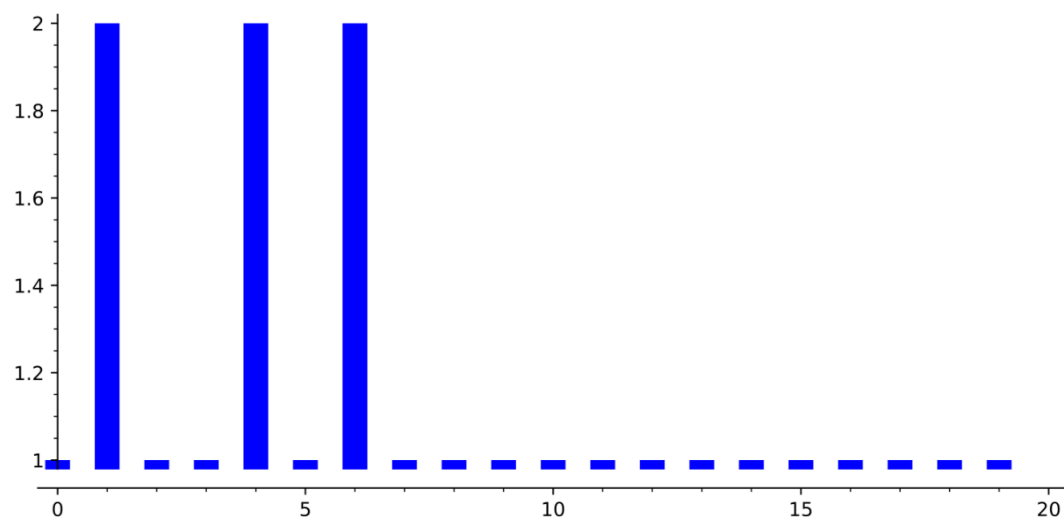
To get a handle on how common certain degree sequences are across graphs of a small number of nodes, we used Sage to compute the degree sequences of all simple graphs up to 9 vertices with none isolated and plot their frequencies in a histogram. Degree sequences were ordered lexicographically to form the bins. With this ordering, the histograms appear to quickly gain a self-similar pattern, and we can see that the maximum counts seem to get large surprisingly quickly.

The maximum number of graphs with n vertices, beginning with $n = 2$, that share a degree sequence forms the sequence*

$$1, 1, 1, 2, 5, 20, 184, 3020, \dots$$

*Pending in the OEIS A308522

On 5 nodes:



On 6 nodes:

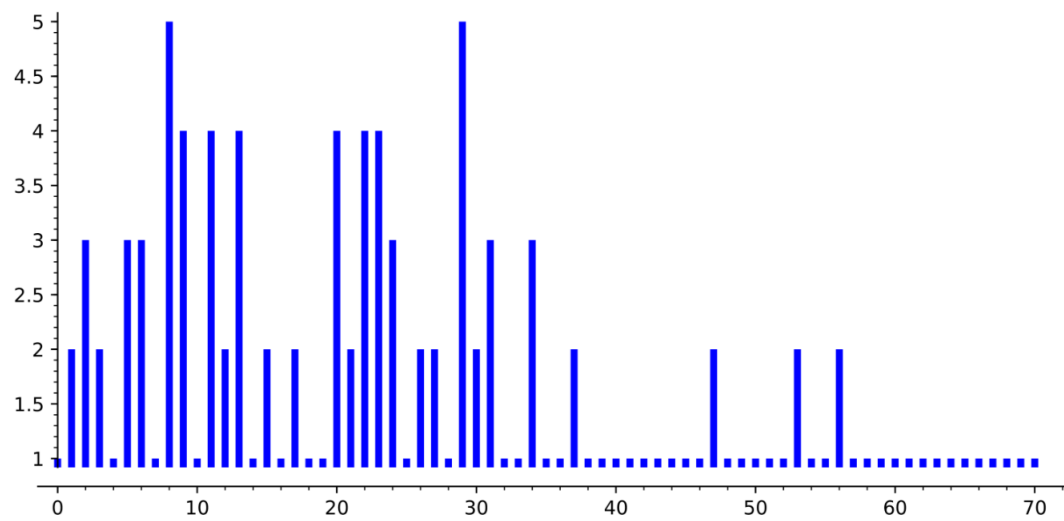
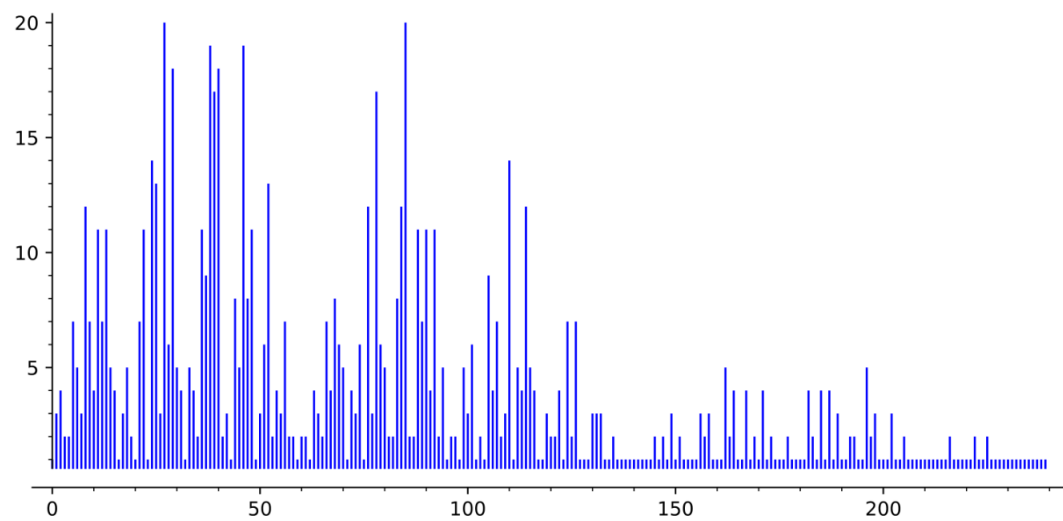


Figure 2.2: Graphs of 5 and 6 Vertices



On 8 nodes:

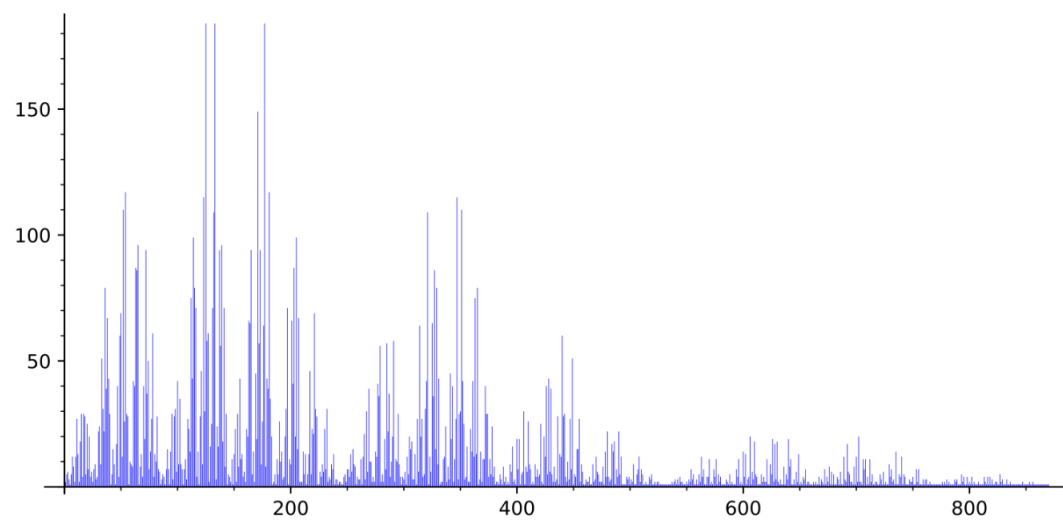


Figure 2.3: Graphs of 7 and 8 Vertices

Histograms

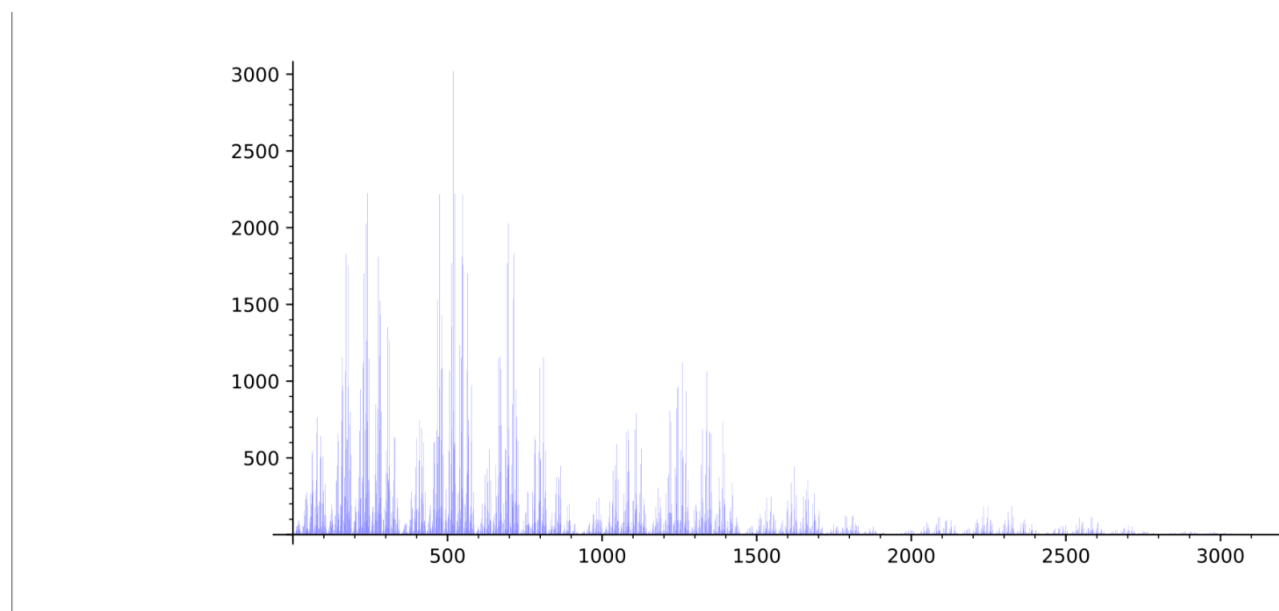


Figure 2.4: Graphs of 9 Vertices

2.3.3 Some Reflections

At first glance, these numbers seem quite high considering we are trying to estimate how many graphs one may have to search through in a given Graphlopedia result before getting to the desired entry. However, this represents the absolute worst case scenario, where all graphs on n vertices are recorded and therefore have been referred to by some citation, as per the requirements of a fingerprint database. While it can be facetiously proven that all graphs are interesting, it is almost assured that not all graphs are interesting enough to be cited in literature. Not all graphs are required or intended to appear in Graphlopedia, just as not all integers show up meaningfully in the OEIS.[9] We therefore must predicate the effectiveness of searching by degree sequence on a criterion being 'interesting', implicitly defined by the mathematical and scientific community at large.

Chapter 3

IMPLEMENTATION DETAILS

3.1 Searching Graphs

Searching for graphs is done primarily through the search box available at the top of any page on *Graphlopedia*.

With the fact that searching by degree sequence may result in manually sifting through graphs in mind, we have taken care to make the search process more efficient in some key instances.

When the exact graph desired is known, one may search the database for the exact graph, either by uploading or writing a Graphlopedia NetworkGraph structured JSON file, or by drawing the graph manually into a canvas. The search will provide an isomorphic graph in the database if available, and otherwise will prompt the user to consider submitting the graph to the database, so others may enjoy the fruit of their labors.

When the exact graph desired is not known but some degrees are known, exact matches to degree sequence queries will always be displayed first. Note that there may be several graphs with the same degree sequence, so some exploration is required in this instance. Matches are sorted by ‘relevance’ - meaning in cases where the query is a subsequence, degree sequences where the query appears earliest in the list are displayed first. This ensures that one need only search for the ‘most significant’, largest degree nodes known in a given graph to be given a high chance of finding a helpful entry. The query need not be consecutive in the search results either.

3.2 Submitting Entries

Submissions to the database can be made on the `\submit` page.

As mentioned earlier, the contribution of the mathematical community at large to this database is critical to its functioning and future relevance. Therefore, it is essential that the barrier to entry for the average user is as low as possible. Unlike many similar databases, we offer the option of

registering for an account on the site - which will allow a user to make updates/edits to existing entries - but we do not require an account to submit an entry.

The submission process, without a user account, is equivalent to the process with an account. All submissions are automatically validated before being added to the database.

3.2.1 Graph Validation

When a user submits a new graph to the database, the following validation processes occur before the graph is accepted:

- Graphlopedia NetworkGraph Schema Validation - The graph is tested against a minimally modified version of the NetJSON NetworkGraph schema which is the preferred format for storing graphs on Graphlopedia. If the graph was drawn in the canvas, this formatting specification is automatically done. Any errors that occur in the process are communicated to the user.
- Non-Emptiness Validation - The graph is tested to be non-empty. Any errors that occur in the process are communicated to the user.

An empty graph is considered to be any graph that contains no edges. Empty graphs include the trivial graph, with no nodes nor edges.

- Consistency Validation - The graph is tested for internal consistency. This means that declared edges never involve undeclared nodes, there is only one edge between any two nodes (multigraphs are as yet unsupported), and that the graph was successfully able to be loaded into a NetworkX object. Any errors that occur in the process are communicated to the user.
- Isomorphism Validation - The graph is tested to see if it is isomorphic to any graph currently in the database. If true, the GID of the isomorphic graph is displayed and the user is prompted to contribute to the isomorphic entry rather than adding an entirely new entry. Currently, only registered users may perform this action.

3.2.2 Reference Validation

Validating references is more difficult than validating graphs, since the content of a given book or paper is significantly harder to determine programmatically. This is primarily where we rely on the mathematical community's expertise in a given subject to provide an accurate page number for the location of a graph in a source. However, although we cannot validate the content of a given reference, we can at very least validate the structure of the citation very easily.

- All string based entries, like titles and authors, are validated before submission and publication dates are validated to represent true calendar dates.
- Any reference submitted *must* include the relevant page number/numbers where a given graph occurs. Generic references to entire books or articles will not be accepted.
- If a reference is provided as an ArXiv Identifier, the paper's title, authors, publication date, and link are automatically loaded into the database
- If a reference is a book or paper journal, one must enter the title, authors, publication date, and page number manually, but formatting will be taken care of upon display.

3.2.3 Security

Since the submission process has been freed up to be publicly available, there are naturally more security concerns involved. To combat this, all forms on *Graphlopedia* are escaped to prevent SQL injections and standard precautions are taken to prevent Cross-Site Request Forgeries (CSRF). This in combination with submission validation should be enough of a buffer to prevent a majority of the database tampering that can occur. In addition, no sensitive information about users is ever stored on the site.

3.2.4 Guest Submissions

As mentioned earlier, one does not need to register with a user account to make submissions to Graphlopedia. However, an unregistered user cannot make edits to any entries that already exist. Currently, unregistered users can neither edit graphs nor the references that are tied to them, but this restriction will be lifted to allow for edits to the list of references from guest submissions

when better precautions are taken to ensure that the system will not be abused or mishandled.

Registering for an account can be done on the `\login` page.

Chapter 4

FUTURE FEATURES

4.1 Searching by Edge Sequence

We plan to soon implement further graph invariant filters, such as filtering by number of vertices, planarity, and other Boolean or numerical invariants. These filtering methods will be applied on top of searching by fingerprint to further narrow searches in an easy and efficient manner.

In addition, we plan to add McKay's canonical labeling to the the graphs that we have currently, and allow searching the database for a canonically labeled graph. In the case when one knows the entire graph, this should make it easier to immediately get to the desired entry, and will make submitting graphs to the database more standard. Outputting canonical labelings of graphs is implemented in Sage, (see http://doc.sagemath.org/html/en/reference/graphs/sage/graphs/generic_graph.html) but needs to be merged into system.

Finally, we wish to implement sourcing citation information from DOIs as well as pre-2007 ArXiv identifiers. At the moment, the automatic ArXiv sourcing functions for the newer identifier format, but has yet to be changed to incorporate the older version. It would also be a nice addition to open URLs to specified page number when a citation is viewed.

Chapter 5

APPENDIX

5.1 Graphlopedia NetworkGraph Schema

```
1 {
2   "$schema": "http://json-schema.org/draft-04/schema#",
3   "type": "object",
4   "additionalProperties": true,
5   "required": [
6     "nodes",
7     "links"
8   ],
9   "properties": {
10    "type": {
11      "type": "string",
12      "enum": [
13        "NetworkGraph"
14      ]
15    },
16    "protocol": {
17      "type": "string"
18    },
19    "version": {
20      "type": "string"
21    },
22    "revision": {
23      "type": "string"
24    },
25    "metric": {
```

```
26     "type": "string"
27   },
28   "router_id": {
29     "type": "string"
30   },
31   "topology_id": {
32     "type": "string"
33   },
34   "label": {
35     "type": "string"
36   },
37   "nodes": {
38     "type": "array",
39     "title": "Nodes",
40     "uniqueItems": true,
41     "additionalItems": true,
42     "items": [
43       {
44         "title": "Node",
45         "type": "object",
46         "additionalProperties": true,
47         "required": [
48           "id"
49         ],
50         "properties": {
51           "id": {
52             "anyOf": [
53               {"type": "string"},
54               {"type": "integer"}
55             ]
56           },
57           "label": {
58             "type": "string"
```

```
59     },
60     "local_addresses": {
61       "type": "array",
62       "additionalItems": true,
63       "uniqueItems": true,
64       "items": {
65         "type": "string"
66       }
67     },
68     "properties": {
69       "type": "object",
70       "additionalProperties": true
71     }
72   }
73 }
74 ]
75 },
76 "links": {
77   "type": "array",
78   "title": "Links",
79   "uniqueItems": true,
80   "additionalItems": true,
81   "items": {
82     "type": "object",
83     "title": "Link",
84     "additionalProperties": true,
85     "required": [
86       "source",
87       "target"
88     ],
89     "properties": {
90       "source": {
91         "anyOf": [
```

```
92         {"type": "string"},
93         {"type": "integer"}
94     ]
95 },
96 "target": {
97     "anyOf": [
98         {"type": "string"},
99         {"type": "integer"}
100     ]
101 },
102 "value": {
103     "type": "number",
104     "default": 1
105 },
106 "value_text": {
107     "type": "string"
108 },
109 "properties": {
110     "type": "object",
111     "additionalProperties": true
112 }
113 }
114 }
115 }
116 }
117 }}
```

BIBLIOGRAPHY

- [1] M. AIGNER AND E. TRIESCH, *Realizability and uniqueness in graphs*, Discrete Mathematics, 136 (1994), p. 320.
- [2] A. BARVINOK AND J. A. HARTIGAN, *The number of graphs and a random graph with a given degree sequence*, arXiv e-prints, (2010), p. arXiv:1003.0356.
- [3] S. BILLEY AND STUDENTS, *Graphlopedia*, 2017. Report for Washington Experimental Mathematics Laboratory, June 2017. Students include Kimberly Bautista, Aaron Bode, W. Riley Casper, Dien Dang, Nicholas Farn, Graham Kelley, Stanley Lai, Adharsh Ranganathan, Michael Trinh, Alex Tsun, Katrina Warner.
- [4] S. BILLEY AND B. TENNER, *Fingerprint databases for theorems*, Notices. Amer. Math. Soc., 60 (2013), pp. 1034–1039.
- [5] G. BRINKMANN, K. COOLSAET, J. GOEDGEBEUR, AND H. MLOT, *House of graphs: A database of interesting graphs*, Discrete Applied Mathematics, 161 (2013), p. 311314.
- [6] S. CHATTERJEE, P. DIACONIS, AND A. SLY, *Random graphs with a given degree sequence*, Ann. Appl. Probab., 21 (2011), pp. 1400–1435.
- [7] S. L. HAKIMI, *On realizability of a set of integers as degrees of the vertices of a linear graph. i*, Journal of the Society for Industrial and Applied Mathematics, 10 (1962), p. 496506.
- [8] V. HAVEL, *Poznámka o existenci konečné graf*, asopis pro psstovn matematiky, 080 (1955), pp. 477–480.
- [9] N. JOHNSTON, *11630 is the first uninteresting number [blog post]*, Nov 2013.
- [10] B. D. MCKAY, *Subgraphs of dense random graphs with specified degrees*, arXiv e-prints, (2010), p. arXiv:1002.3018.
- [11] OEIS FOUNDATION INC., *The On-Line Encyclopedia of Integer Sequences*, 2019. Online. <http://oeis.org>.
- [12] M. RUBEY, C. STUMP, ET AL., *FindStat - The combinatorial statistics database*. <http://www.FindStat.org>. Accessed: 2019.
- [13] B. TENNER, *Database of permutation pattern avoidance*. <https://math.depaul.edu/bridget/patterns.html>. Accessed: 2019.