

Solving Electrical Circuits via Graph Theory

Jan Vrbik

Department of Mathematics, Brock University, St. Catharines, Ontario, Canada

Email: jvrbik@brocku.ca

How to cite this paper: Vrbik, J. (2022) Solving Electrical Circuits via Graph Theory. *Applied Mathematics*, 13, 77-86.
<https://doi.org/10.4236/am.2022.131007>

Received: November 30, 2021

Accepted: January 22, 2022

Published: January 25, 2022

Copyright © 2022 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Solving for currents of an electrical circuit with resistances and batteries has always been the ultimate test of proper understanding of Kirchoff's rules. Yet, it is hardly ever emphasized that a systematic solution of more complex cases requires good understanding of the relevant part of Graph theory. Even though this is usually not covered by Physics' curriculum, it may still be of interest to some teachers and their mathematically inclined students, who may want to learn details of the rigorous approach. The purpose of this article is to provide a concise derivation of a linear set of equations leading to a unique solution of the problem at hand. We also present a simple computer program which builds such a solution for circuits of any textbook size.

Keywords

Kirchoff's Laws, Connected Graph, Incidence Matrix, Spanning Tree, Fundamental Cycles

1. Introduction

In introductory Physics courses, such as [1], students are taught Kirchoff's current and voltage laws (aka junction and loop rules); to demonstrate their proper understanding, they are then asked to solve a given electrical circuit (involving resistors and sources of voltage) for the resulting currents. This article summarizes the classical approach of several books (e.g. [2] and [3]) to develop a systematic way of building the solution by employing the power of graph theory, and provides the corresponding Mathematica program to demonstrate it. There is a multitude of other books and articles on specific aspects of applying graphs to various topics, such as, for example, [4] and [5]—too numerous to provide their exhaustive list.

Before we start, we need to point out that the two fields we are combining in this article (namely Graph Theory and basic Physics) use rather diverging ter-

minology; here we consistently utilize the notation of Graph Theory: Kirchoff's junctions are called *nodes*, his loops become *cycles* (loop itself acquires a different meaning), and connections between junctions are *links*. We start with a few basic definitions and results of the graph theory; more details can be found in [6].

2. Graphs

A **graph** is a collection of vertices (or **nodes**, in our terminology) visually represented by small circles, together with a collection of edges or **links** (drawn as straight-line segments) connecting some pairs of nodes to one another. Nodes joined by a link are called **adjacent**, or being **neighbors** of each other; a link is **incident** to each of its **end** nodes. To study electrical circuits, our graphs are also subject to the following restrictions:

- 1) Two nodes are *not* allowed to be joined by more than one link (when it does happen in a circuit, we show how to deal with it),
- 2) We also do *not* allow **loops**—links joining a node to itself (such part of a circuit would be trivial to solve); these two conditions amount to saying that the graph is **simple**,
- 3) The graph must be connected (see subsequent definition—disconnected circuits would be easy to deal with *separately*); in addition to this, we do not allow links whose removal would result in a disconnected graph, as such links would carry no current (even if they contain a battery), which would again enable us to deal with the resulting two parts individually.

Consider a sequence of k *distinct* nodes, with *consecutive* nodes being *adjacent*; the corresponding $k - 1$ links constitute a **path**, where all nodes along a path are thereby **connected** to each other. When *any* two nodes of a graph are connected by at least one path, the graph itself is called connected. When the first and last node of a path are further joined by an *existing* link, the path becomes a **cycle**. Note that our restrictions imply that every link of a graph is a part of at least one cycle (removing the link must leave the graph connected by Condition 3, so there must be yet *another* path connecting its two end nodes, which thus completes a cycle).

A **tree** is a collection of links (and their incident nodes) which is *connected* and contains *no* cycle.

2.1. Spanning Tree

For a connected graph, this is a tree connecting *all* its nodes (let us denote their number by n); note that there are normally many spanning trees of a graph, each of them having exactly $n - 1$ links. It is always possible to find a spanning tree by the following process: start with an arbitrary link and its two incident nodes, then add a neighbor of one of these nodes (plus the corresponding link) to the existing selection; repeat till there are no nodes left. Until reaching the last node, there must always be at least one such neighbor (otherwise, the original graph

would *not* have been connected). The links *not* used to complete a spanning tree constitute the corresponding **co-tree** with exactly $m - (n - 1)$ links, where m is the total number of *links* in the graph.

Adding any *one* of the co-tree links to the corresponding spanning tree necessarily creates a cycle (since all nodes are already connected by the tree's links, adding a link between any two of them is bound to achieve this). Thus created $m - (n - 1)$ cycles (discarding the links of the spanning tree not involved in creating the cycle) are called **fundamental cycles** (relative to the spanning tree). Note that *any* cycle of the graph can be expressed as a Δ -sum of those fundamental cycles with which the cycle shares a co-tree link (see [7]), where the Δ -sum of two cycles consists of links which belong to either one of the two cycles but *not* both (clearly a commutative and associative operation: the Δ -sum of several cycles keeps only the links common to an *odd* number of these cycles, discarding the rest).

2.2. Directed Graphs

These are essential for describing electrical circuits; this time, every link must be also assigned a direction (visualize it as an arrow pointing from one node to the other). The n by m **incidence matrix** of a directed graph has a row for each of the graph's nodes, and a column for each of its links, while its elements indicate whether the link is incident to the node by having the value of either $+1$ or -1 (depending on whether the link's arrow is pointing *out* or *into* the node, respectively), and the value of 0 when it is not incident. Note that each column of this matrix has always exactly one $+1$ and one -1 , the remaining elements being 0 (every link has exactly two end nodes). In a connected graph, the *only* linear combination of rows of the incidence matrix to result in a zero vector is then the sum of all rows (times an inconsequential constant) since, to achieve a zero sum, coefficients of two rows (aka nodes) must equal to each other whenever the two nodes are adjacent, further implying that they must be identical for *all* connected nodes. The incidence matrix is thus of rank $n - 1$; after deleting any one of its rows, the remaining rows (we denote the resulting matrix \mathbb{K}) are linearly independent. Therefore $\mathbb{K}\mathbf{i} = \mathbf{0}$, where $\mathbf{0}$ is a column vector of $n - 1$ zeros and \mathbf{i} is the vector of the links' currents, thus enforces Kirchoff's law of currents at each node (including the one which was deleted).

It is easy to show that an $n - 1$ by $n - 1$ sub-determinant of the \mathbb{K} matrix is equal to 0 whenever its columns (aka links) contain a cycle. This is because columns of a cycle can be added/subtracted (the sign is chosen so as to follow the cycle in a consistent, one-way direction) to yield a zero vector (as each node is entered and left exactly once, contributing $+1$ and -1 to the row's total), making them linearly *dependent*. On the other hand, an $n - 1$ by $n - 1$ sub-determinant of the \mathbb{K} matrix is equal to either $+1$ or -1 whenever the columns (aka links) contain *no* cycle (note that $n - 1$ links without a cycle constitute a spanning tree), since then there is exactly *one* nonzero term in the usual $(n - 1)!$

term expansion of such a sub-determinant (there is only one way of matching incident nodes to links of a spanning tree, after one of the nodes has been deleted to create matrix \mathbb{K} ; make this node the root of the tree to visualize the unique link \leftrightarrow node correspondence). Since $\det(\mathbb{K}\mathbb{K}^T)$ is equal to the sum of squares of all the $n - 1$ by $n - 1$ sub-determinants of \mathbb{K} (by Cauchy-Binet formula), its value must thus equal to the total number of different spanning trees of the graph.

2.3. Circulation Matrix

Next, using a *specific* spanning tree, we assign each fundamental cycle of a directed graph a *consistent* orientation (of its co-tree link) and define an $m - (n - 1)$ by m **circulation matrix** \mathbb{C} whose elements specify which links (columns of \mathbb{C}) form each such fundamental cycle (represented by a row of \mathbb{C}) by using +1 (when the cycle follows the link's direction) or -1 (when moving against it); 0 implies that the link is not a part of the cycle.

We now show that an $m - (n - 1)$ by $m - (n - 1)$ *sub-determinant* of this matrix equals 0 whenever the selected columns/links contain a **bond**; to define a bond, we separate nodes of a graph into two distinct sets and remove the links which span the divide (having one end on each side); when the nodes of each side remain connected, the removed links constitute a bond. Similarly to fundamental cycles, each bond is given a consistent (from one side of the divide to the other; either choice is good) orientation. Note that the sum/difference (when following/going against, the bond's orientation) of those links (columns) of \mathbb{C} which form a bond results in a zero vector (each fundamental cycle/row, when crossing the divide, must then come back, thus contributing +1 and -1 to the sum), making these columns linearly *dependent*. Also note that any collection of links whose removal would make the graph disconnected must contain a bond.

On the other hand, when the $m - (n - 1)$ removed links do *not* contain a bond (equivalently, when they form one of the many possible co-trees, *i.e.* removing them keeps the graph connected by the remaining $n - 1$ links), the corresponding sub-determinant equals to either +1 or -1. This follows from the following relationship between two circulation matrices (based on two *different* spanning trees and the corresponding co-trees/fundamental cycles)

$$\mathbb{C}_1 = \mathbb{C}_1 |_{S_2} \cdot \mathbb{C}_2 \tag{1}$$

where $\mathbb{C}_1 |_{S_2}$ denotes the \mathbb{C}_1 matrix, reduced to those columns/links which belong to co-tree S_2 ; note that elements of the resulting $m - (n - 1)$ by $m - (n - 1)$ matrix indicate which links of co-tree S_2 are a part of each individual fundamental cycle (row) of \mathbb{C}_1 , and whether their directions agree or are opposite. We then correspondingly add/subtract fundamental cycles/rows of \mathbb{C}_2 to convert them back to fundamental cycles/rows of \mathbb{C}_1 (this is a directed version of the previous Δ -sum algorithm, carried out by the indicated matrix multiplication in (1); whether to add, subtract or ignore a row of \mathbb{C}_2 is determined by its co-tree link (being a part of the \mathbb{C}_1 cycle and having the same di-

rection, opposite direction, or not being a part of it at all, respectively—see [8] [9]).

Keeping only the S_1 columns of each side of (1) and computing the resulting determinant yields

$$\det(\mathbb{C}_1 | S_1) = \det(\mathbb{C}_1 | S_2) \det(\mathbb{C}_2 | S_1) \tag{2}$$

Since the determinant on the left-hand side has only *one* non-zero term (there is only one unique way or matching fundamental cycles to the *corresponding* co-tree links), the determinant’s value is thus either +1 or -1. And since both determinants on the right hand must be integers, +1 or -1 are the only two possibilities for each of them as well.

This proves that $\det(\mathbb{C} | S)^2 = 1$ for any spanning tree \mathbb{C} and any (not necessarily the corresponding) co-tree S ; when it is *not* a co-tree, S must contain a bond (removing links of S makes the graph disconnected), implying that $\det(\mathbb{C} | S) = 0$.

Using the Cauchy-Binet formula, we thus get

$$\det(\mathbb{C}\mathbb{C}^T) = \sum_{\text{All } S} \det(\mathbb{C} | S)^2 \tag{3}$$

where S is an arbitrary selection of $m - (n - 1)$ columns of the \mathbb{C} matrix. The resulting sum thus yields the number of the graph’s co-trees (equivalently, of the graph’s spanning trees).

It is a simple observation that rows of \mathbb{C} and \mathbb{K} are *mutually* orthogonal (when a cycle enters a node, it must also leave it, contributing +1 *and* -1 to the dot product of any such pair of rows). This further implies that the determinant of the combined matrix $\begin{bmatrix} \mathbb{K} \\ \mathbb{C} \end{bmatrix}$ is also equal to the number of spanning trees, but

it may come with a negative sign. The next line shows why

$$\begin{aligned} \left(\det \begin{bmatrix} \mathbb{K} \\ \mathbb{C} \end{bmatrix} \right)^2 &= \det \left(\begin{bmatrix} \mathbb{K} \\ \mathbb{C} \end{bmatrix} \cdot \begin{bmatrix} \mathbb{K}^T & \mathbb{C}^T \end{bmatrix} \right) \\ &= \det \begin{bmatrix} \mathbb{K}\mathbb{K}^T & \mathbb{O} \\ \mathbb{O} & \mathbb{C}\mathbb{C}^T \end{bmatrix} \\ &= \det(\mathbb{K}\mathbb{K}^T) \det(\mathbb{C}\mathbb{C}^T) \end{aligned} \tag{4}$$

This means that each spanning tree (selection of columns of \mathbb{K}) and its co-tree (the corresponding columns of \mathbb{C}) contributes exactly one non-zero term (equal to 1 or -1) to the usual $m!$ term expansion of the determinant of $\begin{bmatrix} \mathbb{K} \\ \mathbb{C} \end{bmatrix}$; these are the *only* nonzero terms of this expansion, and they must all have the same sign.

2.4. Kirchoff Laws

It is easy to see that $\mathbb{C}\mathbb{R}\mathbf{i} = \mathbb{C}\mathbf{v}$ enforces Kirchoff’s voltage law for each fundamental cycle (and thus for *any* cycle, as we show shortly), where \mathbb{R} is a *main-diagonal* matrix of resistances (one for each link) and \mathbf{v} is the corresponding vector of applied voltages. From what we have done so far, it follows that the

determinant of $\begin{bmatrix} \mathbb{K} \\ \text{CR} \end{bmatrix}$ is equal to (up to a sign) the sum over all co-trees of the product of the corresponding resistances. Note that the rank of the last matrix thus remains equal to m whenever at least one term of this sum remains positive (*i.e.* links with zero resistance are a subset of a spanning tree, so that the corresponding co-tree links are all positive). An equivalent condition is that the set of zero-resistance links does not contain a cycle; this will thus ensure that the following m by m system of equations

$$\begin{bmatrix} \mathbb{K} \\ \text{CR} \end{bmatrix} \cdot \mathbf{i} = \begin{bmatrix} \mathbf{0} \\ \text{Cv} \end{bmatrix} \quad (5)$$

has a *unique* solution for \mathbf{i} . Note that allowing zero-resistance links with at least one voltage source form a cycle would result in infinite current (a short circuit); without such voltage source, there is infinitely many finite solutions (the subsequent program returns one of them).

Also note that *any* cycle can be built from fundamental cycles, as we have already shown. Meeting Kirchoff's voltage law for all fundamental cycles then ensures its validity for any cycle.

3. Electrical Circuits

Having thus found a symbolic solution to an electric circuit consisting of *resistors*, *batteries* (idealized sources of voltage with no internal resistance), and connecting *links* (also of negligible resistance), we now convert the algorithm into the corresponding Mathematica program. As input, we need to specify the value of *resistance* in each link (when there is more than one resistor, just add the resistances), and of the total *voltage* of batteries in each link (there is usually only one, if any); note that, unlike resistances, each voltage may have either sign, depending whether its direction agrees (or not) with the link's orientation. The program then yields the resulting current flowing through each link (also of either sign). Rather than using specific units, we just assume that they are matched in the following way: a battery of voltage v , connected (in a closed loop) to a resistor of resistance r , generates a current (in the voltage's direction) of magnitude v/r .

3.1. Mathematica Program

A complete information about any such circuit can be encoded by first labelling all its *nodes* using n distinct symbols, and then providing a list (in any order) of all links, together with the corresponding resistance and voltage. For each link we thus have to enter up to four symbols/numbers, identifying: the incident nodes (their order specifies the link's orientation), the corresponding resistance, and the battery's voltage. When there is no battery, specify only the nodes and resistance; when both the voltage and resistance are zero, the two nodes will suffice.

Thus completed list is then used as an argument of the following Mathematica program:

```

circuit[in_]:=
Module[{inp = Map[PadRight[#, 4] &, in], lk, r, v, gr, K, C, fc},
lk = Map[Take[#, 2] &, inp];
gr = Graph[lk/. {a_, b_} -> a\[DirectedEdge] b];
r = Map[#[[3]] &, inp]; v = Map[#[[4]] &, inp];
K = Drop[Normal[IncidenceMatrix[gr]], 1];
fc = FindFundamentalCycles[UndirectedGraph[gr]];
fc = fc /. a_\[UndirectedEdge] b_ -> {a, b};
C = Outer[Count[#1, #2] &, fc, lk, 1]
  -Outer[Count[#1, #2] &, fc, Map[Reverse, lk], 1];
LinearSolve[Join[K, Map[r # &, C]], Join[Table[0, Length[K]], C.v]]]

```

The program takes the input denoted “in” and extends its items with fewer than four symbols/numbers by padding it with one or two zeros, calling the new list “inp”. Based on “inp”, the next line then creates the corresponding list of m links “lk”, making it into a directed graph “gr”; the remaining part of “inp” is then converted into a vector “r” of the corresponding resistances, and a similar vector “v” of the links’ voltages. The code continues by building the n by m incidence matrix and deleting its last (redundant) row (matrix \mathbb{K}), and a list of $m - n + 1$ fundamental cycles “fc” of graph “gr”; the next two lines convert this list into matrix \mathbb{C} . Each row of \mathbb{C} is then multiplied (*element-wise*) by vector “r”, and the resulting matrix is appended to \mathbb{K} ; similarly, \mathbb{C} is multiplied (this time, it is the regular matrix by vector multiplication) by vector “v” and the result is appended to a vector of $n - 1$ zeros. This is done in the last row of the program, which then also solves the corresponding set of linear equations for the resulting m currents.

3.2. Examples

Consider the following circuit (**Figure 1**), where the boxed numbers specify the links’ resistances and the values in double-square brackets are the applied voltages (note that the $C \rightarrow G$ and $B \rightarrow F$ links have no resistance).

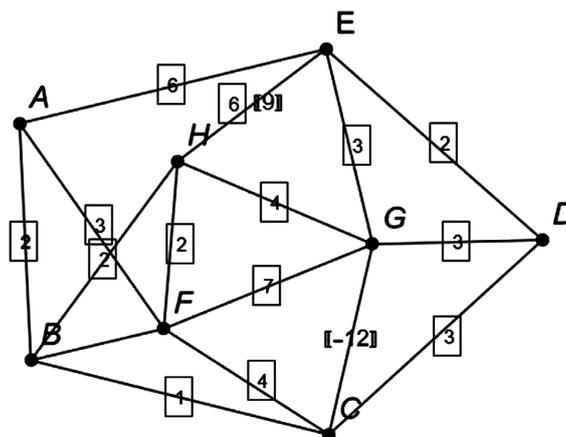


Figure 1. Circuit with 8 nodes, 16 links, 14 resistors and 2 batteries.

Typing:

```
circuit [{{A, B, 2}, {A, E, 6}, {A, F, 3}, {B, C, 1}, {B, F}, {B, H, 2}, {C, D, 3}, {C,
F, 4}, {C, G, 0, -12.}, {D, E, 2}, {D, G, 3}, {E, G, 3}, {E, H, 6, 9}, {F, G, 7}, {F, H, 2},
{G, H, 4}}]
```

then yields the desired solution, namely

```
{-0.515, 0.858, -0.343, -2.777, 1.614, 0.648, 2.422, 0.694, -5.893, 0.844, 1.578,
1.015, 0.686, 1.318, 0.648, -1.982}
```

Note that the corresponding graph is *not planar*, meaning that its two-dimensional rendering always leads to some links crossing each other without physical contact; being planar or not is of great importance in Graph Theory, but rather inconsequential for our treatment of electrical circuits.

When there are multiple links between the same two nodes, each but one of these links must be subdivided into two links (following the same orientation) by inserting an extra node anywhere along the link; in the solution, both links will of course carry the same current and the extra node can be ignored.

As an example, consider the following circuit (**Figure 2**) (the nodes are now labeled by integers)

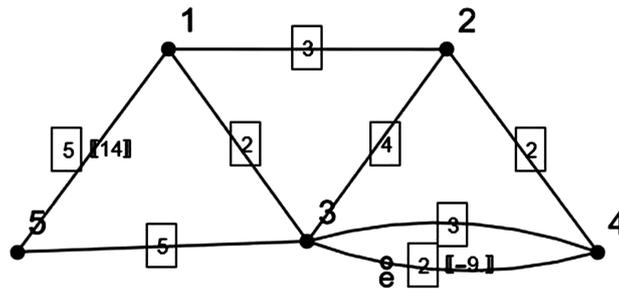


Figure 2. Circuit with double link, removed by inserting an extra node e.

which can be solved by typing

```
circuit [{{1, 2, 3}, {3, 5, 5}, {1, 3, 2}, {1, 5, 5, 14.}, {2, 3, 4}, {2, 4, 2}, {3, 4, 3}, {3,
e}, {e, 4, 2, -9.}}]
```

resulting in the following currents

```
{0.103, -1.149, -1.253, 1.149, -0.704, 0.807, 1.477, -2.284, -2.284}
```

where the last two values are identical (as they must be) and can be merged into a single answer; note that we have placed an extra node “e” inside what would have been (if the program allowed it) a {3, 4, 2, -9} link.

3.3. Extension to Capacitors

The same program can also solve, in a practically identical manner, circuits containing only batteries and capacitors (instead of resistors). The main difference is that, instead of specifying the resistance of each link, we have to replace it by the *reciprocal* of the link’s capacitance, since now the electrical charge stored on each capacitor equals $v \cdot c$, where c is the capacitance. The output then yields the *charge* on each link’s capacitor; when there are two or more capacitors (in

series) in the same link, their reciprocal capacitances need to be added, and the resulting charge then applies to each of them. When a link has no capacitor at all, we must enter 0 for the corresponding reciprocal (think of it as having *infinite* capacitance), and the resulting charge is the one which has gone through the link to charge the rest of the circuit.

As an example, consider the following circuit (**Figure 3**)

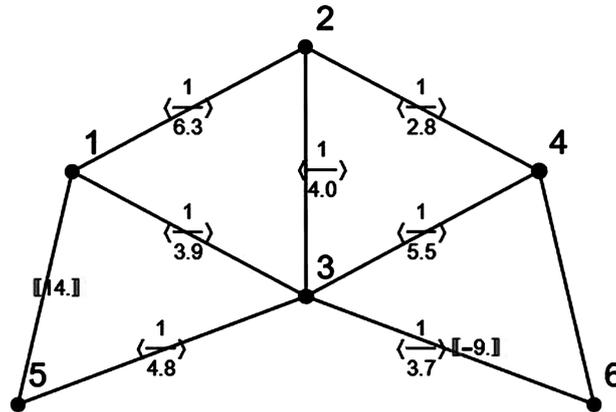


Figure 3. Circuit with 7 capacitors and 2 batteries.

where a pair of brackets stands for a capacitor whose capacitance is specified in the corresponding denominator. The solution is procured by typing

```
circuit [{{1, 2, 1/6.3}, {3, 5, 1/4.8}, {1, 3, 1/3.9}, {1, 5, 0, 14.}, {2, 3, 1/4.}, {2, 4, 1/2.8}, {3, 4, 1/5.5}, {3, 6, 1/3.7}, {6, 4}}]
```

which returns the following charges

```
{-17.701, -39.890, -22.189, 39.890, -11.519, -6.18198, 3.69575, 2.4862, 2.4862}
```

The graph makes it obvious why the last two values must be identical.

4. Conclusion

In this article we have presented a valuable tool for analyzing electrical circuits consisting either of resistors and batteries, or of capacitors and batteries. The corresponding mathematical theory has been developed in full detail, including the proof (missing in existing literature) of sufficient and necessary conditions for the corresponding system of linear equations to have a unique solution. A simple Mathematica program has also been provided to enable the reader to readily find such a solution for any given configuration of nodes, links, resistors (capacitors) and batteries. This makes it a useful tool to facilitate proper understanding of such circuits from both practical and theoretical point of view.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Halliday, D. Resnick, R. and Walker, J. (2008) Fundamentals of Physics (Vol. 2). 8th

Edition, John Wiley & Sons, Hoboken.

- [2] Deo, N. (1974) Graph Theory with Applications to Engineering & Computer Science. Dover Publications, New York.
- [3] Chen, W.-K. (1976) Applied Graph Theory: Graphs and Electrical Networks. 2nd Edition, Elsevier Science, Amsterdam.
- [4] Khalifa, W.R. and Jasim, T.H. (2021) On Study of Some Concepts in Nano Continuity via Graph Theory. *Open Access Library Journal*, **8**, 1-9.
<https://doi.org/10.4236/oalib.1107568>
- [5] Kalil, C., de Castro, M., Silva, D. and Cortez, C. (2021) Applying Graph Theory and Mathematical-Computational Modelling to Study a Neurophysiological Circuit. *Open Journal of Modelling and Simulation*, **9**, 159-171.
<https://doi.org/10.4236/ojmsi.2021.92011>
- [6] Bondy, J.A. and Murty, U.S.R (2008) Graph Theory. Springer, Berlin.
<https://doi.org/10.1007/978-1-84628-970-5>
- [7] <https://math.stackexchange.com/q/3952813>
- [8] <https://mathoverflow.net/q/385726>
- [9] <https://math.stackexchange.com/q/4355930>