Networks and Spanning Trees

Jerry Lodder*

1 Introduction

In 1857 Arthur Cayley (1821–1895) published a paper [9] that introduces the term "tree" to describe the logical branching that occurs when iterating the fundamental process of (partial) differentiation. When discussing the composition of four symbols that involve derivatives, Cayley writes "But without a more convenient notation, it would be difficult to find [their] corresponding expressions . . . . This, however, can be at once effected by means of the analytical forms called trees . . . " [9]. Without defining the term "tree," Cayley has identified a certain structure that occurs today in quite different situations, from networks in computer science to representing efficient delivery routes for transportation.

In the paper "On the Theory of the Analytical Forms Called Trees" [9], Cayley is intrigued enough by this new structure that he proceeds to count trees with certain properties. For him, every tree represents a sequence of derivatives applied in a very specific order, terminating at a final or root term denoted U. Cayley actually uses the word "root" in reference to the point corresponding to U. The remainder of the paper enumerates what today are called "rooted trees." However, in a later paper "A Theorem on Trees" [10], published in 1889, Cayley makes a finer distinction when counting trees, so that no one point is considered as the root, but all points carry fixed labels α, β, γ, etc. The British mathematician counts these trees with fixed labels, arriving at a result that today is called "Cayley's formula." Cayley associates to each labeled tree a polynomial, and proceeds to add all polynomials corresponding to labeled trees with n vertices, arriving at a compelling result that depends on n in a very recognizable pattern. The reader is invited to find this pattern, and perhaps follow in Cayley's footsteps of discovery, by working Exercises (2.4), (2.5), (2.6). After systematically counting labeled trees on six vertices, Cayley writes: "It will be at once seen that the proof for this particular case is applicable for any value whatever of n" [10]. This "proof," however, would require an inverse correspondence between Cayley polynomials and labeled trees, which he does not construct. In fact, most of Cayley's polynomials correspond to several possible trees, as outlined in Exercise (2.8).

A complete proof of Cayley's result is offered from the work of the German mathematician Heinz Prüfer (1896–1934), who develops a quite clever and geometrically appealing method for counting labeled trees. He uses no modern terminology, not even the word "tree" in his work. Probably at the suggestion of Issai Schur (1875–1941), Prüfer phrases his arguments in terms of counting railway networks with certain properties [21]: Given a country with n towns, in how many ways can a railway network be constructed so that

1. the least number of railway segments is used; and

*Mathematical Sciences; Dept. 3MB, Box 30001; New Mexico State University; Las Cruces, NM 88003; jlodder@nmsu.edu.

2. a person can travel from each town to any other town by some sequence of connected segments.

The ideas expressed here, that the least number of railway segments is used, yet travel remains possible between any two towns, are recognized today as properties that characterize such a railway network as a (spanning) tree. Since the towns are fixed, their names (labels) are not interchangeable, and a labeled tree is an excellent model for this problem.

Prüfer wishes to count all railway networks satisfying properties (1) and (2) above, and in doing so, he arrives at a result that agrees with Cayley's formula. Prüfer assigns to each tree a particular symbol based on the point labels (town names). Counting the resulting symbols is then much easier than counting trees. Of course, establishing a one-to-one correspondence between symbols and trees requires some work, which Prüfer writes "follows from an induction argument" (on the number of towns) [21]. By the time of Prüfer's writings, the idea of a one-to-one correspondence was understood from the work of Georg Cantor (1845–1918), and mathematical induction was a well-developed argument form. A useful exercise is the comparison of Prüfer's use of symbols and Cayley's use of polynomials for the purpose of counting the same structures (Exercises 2.4, 3.14). As a result, we know that there are n^{n-2} possible labeled trees that can be formed from n towns.

In 1926 Otakar Borůvka (1899–1995) published [2, 3] the solution to an applied problem of immediate benefit for constructing an electrical power network in the Southern Moravia Region, now part of the Czech Republic. In recalling his own work, Borůvka writes [4, 12]:

> My studies at polytechnical schools made me feel very close to engineering sciences and made me fully appreciate technical and other applications of mathematics. Soon after the end of World War I, at the beginnings of the 1920s, the Electrical Power Company of Western Moravia, Brno, was engaged in rural electrification of Southern Moravia. In the framework of my friendly relations with some of their employees, I was asked to solve, from a mathematical standpoint, the question of the most economical construction of an electric power network. I succeeded in finding a construction ... which I published in 1926 ... .

Let's examine specifically how Borůvka phrased the problem [3]:

> There are n points in the plane (in space) whose mutual distances are all different. We wish to join them by a net such that:
>
> 1. Any two points are joined either directly or by means of some other points.
> 2. The total length of the net would be the shortest possible.

How does this problem differ from that posed by Prüfer? Prüfer wishes to find a network that requires the least number of single segments, while Borůvka wishes to find a network of shortest possible total length. Both authors require that all towns in their respective applications be connected to the network (railway or electrical). Are these identical problems? No, since Prüfer never considers the length of a railway segment connecting two towns. Are these problems related? Yes, since a network of shortest total length is recognized today as a tree (Exercise 4.1). Thus, of all possible n^{n-2} labeled trees on n points (towns), which tree or trees have the shortest possible total length? Borůvka offers a solution to this problem that is rather algorithmic in nature, and has become the basis for finding what today is called a *minimum spanning tree*. The Czech mathematician, however, uses no modern terminology in his 1926 papers, not even the word "tree."

Since the writings of Prüfer and Borůvka, an entire field of study has arisen to provide a framework for discussing these and similar problems in network design. This is the field of graph theory, and a tree is recognized today as a graph with certain properties. Modern mathematics offers a host of lemmas and theorems about trees, many of which reflect observations made in these earlier writings. The reader is asked to be guided by inquiry, experiment, and discovery as we explore "Networks and Spanning Trees" from the works of the pioneers.

2 Cayley's Analytical Forms Called Trees

Arthur Cayley (1821–1895) was a prolific scholar, publishing over 1,000 articles in various fields of mathematics, and refereeing hundreds of others. He studied at Trinity College in Cambridge, England. Upon graduation, he worked as a lawyer, although he pursued mathematics in his spare time. In 1863 he became Sadlerin Professor of Mathematics at Cambridge [13]. We examine only two of Cayley's papers on trees [9, 10], with his pioneering work on group theory developed in a separate monograph, "Abstract Awakenings in Algebra: Early Group Theory in the Works of Lagrange, Cauchy and Cayley," [1].

In an 1857 publication [9], Cayley introduces the term "tree" to describe the logical branching that occurs when iterating the fundamental procedure of differentiation. Calculus is not the main concern here, but instead an organizational tool is developed that provides a visual overview of the individual terms under differentiation. This tool is used today in quite different situations from networks in computer science to finding efficient delivery routes in the transportation industry. To introduce Cayley's paper, let ∂_x denote differentiation with respect to x and let ∂_y denote differentiation with respect to y. Then $\partial_x(x^2y) = 2xy$ and $\partial_y(x^2y) = x^2$, given that x and y are independent variables. The symbols ∂_x and ∂_y are called operators, while the expression x^2y itself is an operand. Note that the symbols ∂_x and ∂_y are applied to (operate on) functions written to the right of the symbol. If the function is written to the left, such as $x^2y\partial_x$, Cayley dubs the entire expression an operandator, and x^2y remains unaltered by the operator on the right. Cayley wishes to study how operandators interact among themselves. Let $P = x^2y\partial_x$ and $Q = xy\partial_y$. Then PQ is the operandator given by

$$PQ = x^2y\partial_x(xy\partial_y) = x^2y(y\partial_y) = x^2y^2\partial_y\,,$$

and QP is the operandator given by

$$QP = xy\partial_y(x^2y\partial_x) = xy(x^2\partial_x) = x^3y\partial_x\,.$$

In the above example, $QP \neq PQ$.

For operandators Q, P, U, what should be the meaning of QPU? Do the groupings $Q(PU)$ and $(QP)U$ yield the same result, with Q, P and U are in the same relative order? (See Exercise (2.1).) First, let's read Cayley's analysis [9] of these questions, and his theory for an efficient method of representing iterated applications of operandators such as $RQPU$.

On the Theory of the Analytical Forms Called Trees.

A symbol such as $A\partial_x + B\partial_y + \ldots$, where A, B, &c. contain the variables x, y, &c. in respect to which the differentiations are to be performed, partakes of the natures of an operand and operator, and may be therefore called an Operandator. Let P, Q, R, ... be any operandators, and let U be a symbol of the same kind, or to fix the ideas, a mere operand; PU denotes the result of the operation P performed on U, and QPU denotes the result of the operation Q performed on PU; and generally in such combinations of symbols, each operation is considered as affecting the operand denoted by means of all the symbols on the right of the operation in question. Now considering the expression QPU, it is easy to see that we may write

$$QPU = (Q \times P)U + (QP)U,$$

where on the right-hand side $(Q \times P)$ and (QP) signify as follows: viz. $Q \times P$ denotes the mere algebraical product of Q and P, while QP (consistently with the general notation as before explained) denotes the result of the operation Q performed upon P as operand; and the two parts $(Q \times P)U$ and $(QP)U$ denote respectively the results of the operations $(Q \times P)$ and (QP) performed each of them upon U as operand. It is proper to remark that $(Q \times P)$ and $(P \times Q)$ have precisely the same meaning, and the symbol may be written in either form indifferently. But without a more convenient notation, it would be difficult to find the corresponding expressions for $RQPU$, &c. This, however, can be at once effected by means of the analytical forms called trees (see figs. 1, 2, 3) which contain all the trees which can be formed with one branch, two branches, and three branches respectively.

The inspection of these figures will at once show what is meant by the term in question, and by the terms *root*, *branches* ..., and *knots* .... To apply this to the question in hand, PU consists of a single term represented by fig. 1 (*bis*); QPU consists, as above, of two terms represented by the two parts of fig. 2 (*bis*), viz. the first part represents the term $(Q \times P)U$, and the second part represents the term $(QP)U$. ...

Fig. 1.

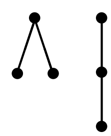


Fig. 2.

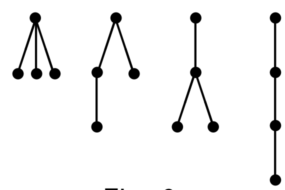


Fig. 3.

Fig. 1 (*bis*).

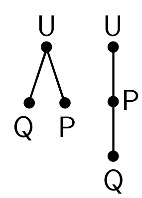


Fig. 2 (*bis*).

Using Cayley's notation, we let A and B denote operands, i.e., functions[1] of the variables x and

[1] infinitely differentiable

y. Let $P = A\partial_x$, $Q = B\partial_y$, and suppose that U is a "mere operand." To ease notation, let

$$\begin{aligned} A_x &= \frac{\partial A}{\partial x}, \qquad B_x = \frac{\partial B}{\partial x}, \qquad U_x = \frac{\partial U}{\partial x}, \\ A_y &= \frac{\partial A}{\partial y}, \qquad B_y = \frac{\partial B}{\partial y}, \qquad U_y = \frac{\partial U}{\partial y}, \\ A_{xy} &= (A_x)_y = \frac{\partial}{\partial y}\left(\frac{\partial A}{\partial x}\right), \ \ldots \ . \end{aligned}$$

Then $PU = A\partial_x(U) = AU_x$, where the latter is the product of the functions A and U_x. (See Exercise 2.1.) Also,

$$(QP)U = (B\partial_y(A\partial_x))(U) = (BA_y\partial_x)(U) = BA_yU_x.$$

Following Cayley, we read "QPU denotes the result of the operation Q performed on PU." Thus, QPU would today be written as $Q(PU)$. In general, is $Q(PU) = (QP)U$? (See Exercise 2.1.) To compute

$$Q(PU) = B\partial_y(AU_x)$$

requires the product rule for differentiation. We have

$$Q(PU) = BAU_{xy} + BA_yU_x.$$

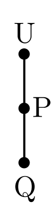
Certainly BA_yU_x matches $(QP)U$ above. How is $(Q \times P)U$ to be interpreted if

$$QPU = (Q \times P)U + (QP)U \ ?$$

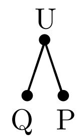
Cayley states that "$Q \times P$ denotes the mere algebraical product of Q and P." This suggests that the correct interpretation of $Q \times P$ is:

$$\begin{aligned} (Q \times P)U &= (B\partial_y \times A\partial_x)(U) \\ &= (BA\partial_y\partial_x)(U) \\ &= (BA\partial_y)(U_x) \\ &= \ BAU_{xy}. \end{aligned}$$

The British mathematician introduces a structural device, called a "tree," to display the various terms needed to represent QPU. First $(QP)U$ is denoted by the tree



and $(Q \times P)U$ is denoted by



In Exercise 2.2 we explore the meaning of the terms "branches," "knots," "root," and study the use of branching as an organizational tool for differentiation. The same tool can be used to represent logical branching in many other circumstances.

Exercise 2.1. Let $A = A(x, y) = x^2y$, $B = B(x, y) = xy^2$, $P = A\partial_x$, $Q = B\partial_y$, and $U = U(x, y) = x + xy$.

(a) Compute PU in terms of the variables x and y.

(b) Compute $U_x = \frac{\partial U}{\partial x}$ and separately compute AU_x.

(c) Compare the product AU_x with PU.

(d) Compute $(QP)U$ in terms of the variables x and y.

(e) Compute $Q(PU)$ in terms of the variables x and y.

(f) Is $(QP)U = Q(PU)$? Justify your answer using parts (d) and (e).

Exercise 2.2. Cayley claims that the terms "root," "branches," and "knots" are clear by inspection.

(a) In the tree representing $(Q \times P)U$, what are most likely the branches? the knots? the root? Keep in mind that $(Q\times P)U$ yields the same result as $(P\times Q)U$, so there is little to distinguish either P or Q as a root. How many branches are connected to the root? What is the effect of each branch that is connected to U?

(b) In the tree representing $(QP)U$, what are the branches? the knots? the root? What is the effect of connecting one knot to another in terms of differentiation?

Exercise 2.3. Let A, B, C, U be operands, i.e., functions[2] of the independent variables x, y, z, and let R, Q, P be the operandators given by

$$R = A\partial_x, \quad Q = B\partial_y, \quad P = C\partial_z\,,$$

where x, y, z are independent variables.

(a) Compute $RQPU = R(Q(PU))$ as a sum of six terms, using the product rule where necessary.

(b) For each of the six terms comprising $R(Q(PU))$, find the corresponding tree that represents the term. Be sure to label the knots using the letters R, Q, P, U. Also, justify your answer.

(c) Which trees in part (b) occur in Cayley's figure 3? Are any trees used more than once? which ones?

[2] infinitely differentiable

(d) What trees are needed to represent the terms of $(R(QP))U$? Be sure to justify your answer. Hint: Write $(R(QP))U$ as sum of terms, and find the corresponding tree in Cayley's Fig. 3 for each term.

(e) Keeping the letters R, Q, P, U in the same relative order, how many different ways are there to parenthesize $R\ Q\ P\ U$? Write each of the different parenthesizations.

Cayley continues his paper "On the Theory of Analytical Forms Called Trees" [9] with an enumeration (counting) of trees with n knots. How trees are counted, which trees are counted as different, and which are considered the same, depends on what structures of a tree are being studied. To begin a more detailed study of trees, a knot is today called a vertex and a branch connecting two knots is called an edge. The number of edges connected to a vertex is called the degree of the vertex. For example in the tree representing $(Q \times P)U$ (see Exercise 2.2), the vertex U has degree two, while Q and P each have degree one. Furthermore, in the two trees shown in Cayley's figure 2 above, each has one vertex of degree two, and two vertices of degree one. Should these trees be counted the same, since the left-hand tree of figure 2 is simply the right-hand tree bent in the middle? Cayley would argue no, since in the left-hand tree (figure 2 (bis)), the root U has degree two, while in the right-hand tree, the root U has degree one. Cayley counts what today would be called the number of rooted trees with n vertices.

However, in a later paper published in 1889, "A Theorem on Trees" [10], Cayley makes an even finer distinction in counting. Consider trees with three fixed vertices labeled α, β, γ as follows:

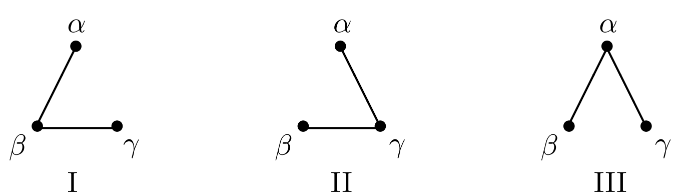


Figure 2.1

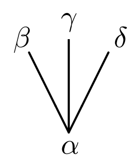
Suppose that the vertices and edges represent physical objects, such as electrical devices connected by wires or towns connected by railway lines. In tree I above, the vertices α and γ are not directly connected by an edge, while in tree II, α and β are not not directly connected, and in tree III, β and γ are not directly connected. Should these three trees be counted as distinct? Cayley does so in his 1889 paper [10], and introduces a method of counting based on assigning polynomials to trees. To motivate the discussion, let's construct polynomials for the above trees by multiplying all pairs of vertices in the given tree that are directly connected by an edge. For tree I, the Cayley polynomial is $(\alpha\beta)(\beta\gamma) = \beta(\alpha\beta\gamma)$. For tree II, the Cayley polynomial is $(\alpha\gamma)(\gamma\beta) = \gamma(\alpha\beta\gamma)$. For tree III, we have $(\beta\alpha)(\alpha\gamma) = \alpha(\alpha\beta\gamma)$. Thus, each polynomial contains the factor $\alpha\beta\gamma$, and one other term for the vertex of degree two. All possible trees on these three vertices are represented by

$$(\alpha + \beta + \gamma)(\alpha\beta\gamma).$$

How many possible trees are there on four fixed vertices α, β, γ, δ ? (See Exercise 2.4.) Cayley denotes the number of vertices (knots) by $n+1$. To count trees on four vertices, use $n+1 = 4$ and $n = 3$ when reading the following excerpt. Without the modern definition of a tree, Cayley writes [10]:

A Theorem on Trees.

The number of trees which can be formed with $n+1$ given knots α, β, γ, ... is $= (n+1)^{n-1}$; for instance $n=3$, the number of trees with the 4 given knots α, β, γ, δ is $4^2 = 16$,

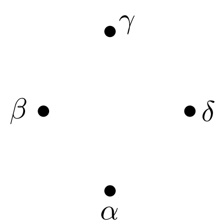


$12 + 4 = 16$

for in the first form ... the α, β, γ, δ may be arranged in 12 different order, ... and in the second form any one of the 4 knots α, β, γ, δ may be in the place occupied by the α: the whole number is thus $12 + 4, = 16$. ...

I use for any tree whatever the following notation: for instance in the first of the forms ... the branches are $\alpha\beta$, $\beta\gamma$, $\gamma\delta$; and the tree is said to be $\alpha\beta^2\gamma^2\delta$ (viz. the knots α, δ occur each once, but β, γ each twice); similarly in the second of the same forms the branches are $\alpha\beta$, $\alpha\gamma$, $\alpha\delta$, and the tree is said to be $\alpha^3\beta\gamma\delta$ (viz. the knot α occurs three times, and the knots β, γ, δ each once). ...

Exercise 2.4. Arrange the four vertices α, β, γ and δ in a fixed configuration, such as the diamond below:



Two trees are considered the same if and only if the same pairs of vertices are directly connected by an edge.

(a) Find all trees in which one knot occurs three times, and three other knots occur once. One of these is the tree that contains "the branches $\alpha\beta$, $\alpha\gamma$, $\alpha\delta$."

(b) For each of the trees in part (a), find the corresponding Cayley polynomial, following the example:

$$(\alpha\beta)(\alpha\gamma)(\alpha\delta) = \alpha^3\beta\gamma\delta.$$

(c) How do the results in parts (a) and (b) compare to Cayley's statement that "any one of the 4 knots may be in the place occupied by the α" ?

(d) Find all trees in which two knots occur once and two knots occur twice. One of these is the tree with branches $\alpha\beta$, $\beta\gamma$, $\gamma\delta$.

(e) Find the Cayley polynomial for each tree in part (d), following the example:

$$(\alpha\beta)(\beta\gamma)(\gamma\delta) = \alpha\beta^2\gamma^2\delta.$$

(f) How do the results in parts (d) and (e) compare to Cayley's statement "the α, β, γ, δ may be arranged in 12 different orders" ?

(g) Add all Cayley polynomials in parts (b) and (e), and compare the result to

$$(\alpha + \beta + \gamma + \delta)^2(\alpha\beta\gamma\delta).$$

(h) Find the sum of all the coefficients in the expansion $(\alpha + \beta + \gamma + \delta)^2$, and compare this to the total number of trees on the fixed vertices α, β, γ, δ.

Exercise 2.5. Following Cayley's example, devise a method for counting all trees on five fixed vertices α, β, γ, δ, ϵ. Be sure to explain your work.

Exercise 2.6. In "A Theorem on Trees" [10], Cayley states that the number of trees on the vertices α, β, γ, δ, ϵ, ζ is equal to the number of terms in the expansion

$$(\alpha + \beta + \gamma + \delta + \epsilon + \zeta)^4(\alpha\beta\gamma\delta\epsilon\zeta).$$

Find the sum of all coefficients in the expansion $(\alpha + \beta + \gamma + \delta + \epsilon + \zeta)^4$, and justify your answer. (Hint: for the purpose of counting, can we set $\alpha = 1$, $\beta = 1$, $\ldots$, $\zeta = 1$?)

After counting the number of trees on six fixed vertices α, β, γ, δ, ϵ, ζ (Exercise 2.6), Cayley simply states "It will be at once seen that the proof given for this particular case is applicable for any value whatever of n" [10]. Presumably, every tree on the $n+1$ vertices x_1, x_2, $\ldots$, x_{n+1} corresponds to exactly one term in the expansion

$$(x_1 + x_2 + x_3 + \ldots + x_{n+1})^{n-1}(x_1x_2x_3 \ldots x_{n+1}),$$

and every term in this expansion corresponds to exactly one tree. Cayley does not state how this correspondence is constructed. For example, when $n = 6$, what tree corresponds to the mononomial

$$(x_1^3x_2^2)(x_1x_2x_3x_4x_5x_6x_7) = x_1^4x_2^3x_3x_4x_5x_6x_7 \text{ ?}$$

Certainly the vertex x_1 will have degree 4 and x_2 will have degree 3. One possibility is the tree with edges given by the pairs

$$(x_1x_2)(x_1x_3)(x_1x_4)(x_1x_5)(x_2x_6)(x_2x_7),$$

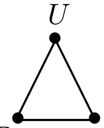
while another possibility is the tree with edges

$$(x_1x_2)(x_1x_3)(x_1x_6)(x_1x_7)(x_2x_4)(x_2x_5).$$

(See Exercise 2.8.) The commutativity of polynomials ($x_ix_j = x_jx_i$) loses information about how the tree is constructed from its list of edges. In the next section, we examine Heinz Prüfer's (1896–1934) method of counting trees with fixed vertices.

First we introduce the term "graph" to describe a figure that can be formed with vertices and edges. Initially used by James J. Sylvester (1814–1897) in a paper entitled "Chemistry and Algebra" [23], the term graph has acquired a rather technical meaning today. Specifically, a *graph* consists of a finite set of vertices $V = \{v_1,\ v_2,\ v_3,\ \dots,\ v_n\}$ and a finite set of edges $E = \{e_1,\ e_2,\ e_3,\ \dots,\ e_k\}$, so that each edge e_i has a starting vertex $u \in V$ and an ending vertex $w \in V$. The starting vertex u could be the same as the ending vertex w for some edge e_i, in which case e_i is called a *loop*. We shall have little reason to consider graphs with loops. Also, there is flexibility in choosing u or w as the starting vertex of an edge, so that an edge connecting u and w may be viewed as an edge connecting w and u.[3] Today graph theory is a rich subject, and our study of trees will lead to special properties of graphs. Certainly every tree is a graph. Is every graph a tree?

What exactly is meant by the term "tree"? Would the following graph in which P, Q and U are operandators qualify as a tree?



If so, what would the meaning of this diagram be in terms of differentiation? Is P applied to Q? or is Q applied to P? To motivate the modern definition of a tree, let's first examine how the idea of a "closed figure" arose in the work of physicist Gustav R. Kirchhoff (1824–1887) in finding the strength of electrical currents in a network of wires. Read the following excerpt [14] [15] for general properties of the network described, and not for details about electricity.

On the solution of the Equations Obtained from the Investigation of the Linear Distribution of Galvanic Current

G.R. Kirchhoff

If we are given a system of n wires 1, 2, ..., n, which are joined to one another in an arbitrary way, ..., then the number of equations necessary for determining the strengths of the currents I_1, I_2, ..., I_n flowing through the wires is obtained by ...

I. If the wires k_1, k_2, ... form a closed figure, and if w_k denotes the resistance of the wire k, and E_k denotes the electromotive force[4], ..., then when I_{k_1}, I_{k_2}, ... are all considered as positive in the *same* direction:

$$w_{k_1} I_{k_1} + w_{k_2} I_{k_2} + \dots = E_{k_1} + E_{k_2} + \dots \ .$$

...

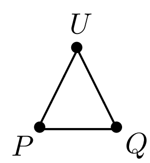
Assuming that the given system of wires does not decompose into quite separate parts, I shall now prove that the solutions of the equations, which are obtained for I_1, I_2, ..., I_n, ..., can be stated in general as follows: ... .

[3] In modern mathematics, this would be called an *undirected graph*.

[4] voltage

Let μ be the least number of wires that must be removed from an arbitrary system so that all the closed figures are destroyed; then μ is also the number of independent equations which can be obtained by using Theorem I. ...

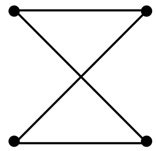
Kirchhoff's description of a wiring diagram is an example of a graph. The vertices are given by the points where two (or more) wires meet, while the edges are those segments of wires that connect meeting points. For each closed figure in the graph, Kirchhoff writes an equation for the currents in the edges of the closed figure. The number of equations needed for Kirchhoff's Theorem I is given by the least number of wires that must be removed from the network so that all the closed figures are destroyed. Key concepts here are "closed figure" and graphs that contain no closed figures. Today a closed figure in a graph is called a *circuit* or a *cycle*, and is defined as a sequence of (distinct) edges e_1, e_2, ..., e_m, such that the ending vertex of e_1 is the starting vertex of e_2, the ending vertex of e_2 is the starting vertex e_3, ... the ending vertex of e_{m-1} is the starting vertex of e_m, and the ending vertex of e_m is the starting vertex of e_1. For example, in the graph



let e_1 be the edge connecting P to Q, e_2 the edge connecting Q to U, and e_3 the edge connecting U to P. Then the sequence of edges e_1, e_2, e_3 forms a circuit.

After discussing closed figures, Kirchhoff states "that the given system of wires does not decompose into quite separate parts," [15] which is an intuitive expression of a *connected graph*. First, a *path* between two vertices u and w is sequence of (distinct) edges e_1, e_2, ..., e_m, such that the starting vertex of e_1 is u, the ending vertex of e_1 is the starting vertex of e_2, the ending vertex of e_2 is the starting vertex e_3, ... the ending vertex of e_{m-1} is the starting vertex of e_m, and the ending vertex of e_m is w. Note that a path from u to w with the additional property that $u = w$ is a circuit, as defined above. A graph G is *connected*, if given any two distinct vertices u and w of G, there is some path between u and w. Kirchhoff's statement "that the given system of wires does not decompose into quite separate parts" is equivalent to supposing that the wiring diagram forms a connected graph. Oswald Veblen (1880–1960), however, was motivated by ideas from the emerging field of topology when he articulated the modern definition of a *tree* as "a connected linear graph which contains no 1-circuits," where a "linear graph" is simply a "graph" in our terminology, and a "1-circuit" is simply a "circuit." Veblen outlined his work on combinatorial topology and graph theory in a series of lectures to the American Mathematical Society in 1916, which appeared as a manuscript in 1922 [24]. Note that having no circuits excludes the possibility of the graph having a loop, i.e., an edge with the same starting vertex and ending vertex. Having no circuits also excludes the possibility of two edges connecting the same pair of vertices, etc.

Exercise 2.7. Using Veblen's definition of a tree as a connected graph that contains no circuits, decide which of the following are trees:

(a) 

(b) The graph with vertices $\{i \in \mathbf{Z} \mid 1 \leq i \leq 10\}$ and edges e_i connecting 1 to i for $i = 2, 3, 4, 5$, and edges e_j connecting 6 to j for $j = 7, 8, 9, 10$.

(c) The graph with vertices $\{i \in \mathbf{Z} \mid 1 \leq i \leq 5\}$ and edges e_i connecting i to $i + 1$, $i = 1, 2, 3, 4$.

(d) The graph with vertices $\{i \in \mathbf{Z} \mid 1 \leq i \leq 5\}$ and edges e_{ij} connecting i to j for $1 \leq i < j \leq 5$.

(f) The graph with vertices given by the students in this class, and edges connecting student A with student B, if A and B have taken some class together before.

Exercise 2.8. Consider all possible trees on the fixed vertices

$$V = \{x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7\}$$

that correspond to the Cayley polynomial $x_1^4 x_2^3 x_3 x_4 x_5 x_6 x_7$.

(a) Sketch the tree with edges connecting the pairs of vertices:

$$(x_1x_2)(x_1x_3)(x_1x_4)(x_1x_5)(x_2x_6)(x_2x_7).$$

(b) Sketch the tree with edges connecting the pairs of vertices:

$$(x_1x_2)(x_1x_3)(x_1x_6)(x_1x_7)(x_2x_4)(x_2x_5).$$

(c) Sketch a tree different from those in (a) and (b) that corresponds to the Cayley polynomial

$$x_1^4 x_2^3 x_3 x_4 x_5 x_6 x_7.$$

(d) Arrange the vertices $x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7$ in a fixed configuration (such as around a circle). Counting two trees as the same if and only if the same pairs of vertices are directly connected by an edge, how many trees correspond to the Cayley polynomial:

$$(x_1^3 x_2^2)(x_1 x_2 x_3 x_4 x_5 x_6 x_7) = x_1^4 x_2^3 x_3 x_4 x_5 x_6 x_7\ ?$$

(e) Sketch each tree in part (d) corresponding to the polynomial $x_1^4 x_2^3 x_3 x_4 x_5 x_6 x_7$.

3 Prüfer's Enumeration of Trees

Heinz Prüfer (1896–1934) is perhaps best known for his contributions to group theory, a topic pioneered by Arthur Cayley and explored in the monograph "Abstract Awakenings in Algebra: Early Group Theory in the Works of Lagrange, Cauchy and Cayley" [1]. Prüfer studied at the University of Berlin under Professors Ferdinand Frobenius and the highly influential Issai Schur. Prüfer writes of his method for counting trees: "I shall express it in an intuitive geometrical garb, as posed by Herr Professor Schur in a problem to the University of Berlin's mathematical seminar" [20, 21]. After receiving his doctorate for his work in group theory, Prüfer accepted an assistantship at the University of Hamburg. In 1927 he became a dozent (lecturer) at the University of Münster, where he remained until his untimely death from lung cancer at the age of 37 [22]. Additionally Prüfer published works on number theory and knot theory, while his lecture notes on projective geometry appeared posthumously.

The number of trees on n fixed (labeled) vertices has become known as "Cayley's formula," although Cayley's exposition is incomplete. (See Exercise 2.8.) His notation $\alpha\beta$ for an edge connecting the vertices α and β can be interpreted as a "transposition," namely an operation whereby α and β are simply switched, perhaps since the tree $\alpha \bullet\!\!-\!\!\!-\!\!\bullet \beta$ is considered the same as $\beta \bullet\!\!-\!\!\bullet \alpha$ for Cayley's purpose of counting trees. Transpositions have become the building blocks of a larger theory known as permutations. In a 1917 publication "Eine Formel der Substitutionstheorie" ("A Formula in Substitution Theory") [11], Berlin Professor Otto Dziobek attempts another proof of Cayley's formula by counting certain permutations that can be constructed from transpositions. As Prüfer writes, Herr Dziobek's proof "is not particularly simple" [21]. Furthermore, counting trees via permutations appears to be a false start. Nonetheless Prüfer uses the term "permutation" in the title of his 1918 paper "A New Proof of a Theorem about Permutations" [20, 21].

Prüfer uses no technical vocabulary to describe graphs or trees, although Veblen's definition of a tree as a connected graph containing no circuits had been stated, at least in lecture, in 1916. Instead, Prüfer introduces the problem via an application: "Consider a country with n towns. These towns must be connected by a railway network of $n-1$ single segments (the smallest possible number) in such a way that one can travel from each town to every other town" [21]. The reader should first identify this as a problem in graph theory. The railway network is a graph with vertices given by the towns and edges given by the "single segments"[5] that directly connect two towns. Prüfer wishes to count all possible railway networks having two salient properties:

(1) the least number of railway segments is used; and

(2) a person can travel from each town to any other town by some sequence of connected segments.

The second property (2) above is tantamount to stating that the graph is connected, while graphs with property (1) are today called minimally connected. A result from modern mathematics is that a connected, minimally connected graph is equivalent to a tree (Exercise 3.4).

Additionally Prüfer states, without justification, that $n-1$ is the least number of railway segments required to produce a network connecting n towns satisfying properties (1) and (2). Is this true no matter what configuration the network (tree) may have? Does every tree on n vertices have exactly $n-1$ edges? These questions are explored in Exercises 3.5 and 3.6. Prüfer continues to identify basic properties of trees via the railway network problem. The statement "The towns at which only one segment terminates we call endpoints" can today be identified as vertices of degree one, and are called *leaves* in modern terminology. Herr Prüfer maintains that the railway networks

[5] "Einzelstreke" in the original German [20].

under consideration always have endpoints. This has become the modern theorem: Every tree has at least one leaf. (See Exercises 3.5 and 3.7.) In fact, every tree with more than one vertex contains at least two leaves (Exercise 3.8).

Let us now read from the original paper [20, 21].

A New Proof of a Theorem about Permutations.

by Heinz Prüfer from Berlin.

In the Berlin Mathematical Society, Herr Dziobek has announced a theorem . . . . His proof . . . is not particularly simple, and it is perhaps of interest to look at another proof which depends entirely on combinatorial considerations. I shall express it in an intuitive geometrical garb, as posed by Herr Professor Schur in a problem to the University of Berlin's mathematical seminar:

Consider a country with n towns. These towns must be connected by a railway network of $n-1$ single segments (the smallest possible number) in such a way that one can travel from each town to every other town. There are n^{n-2} different railway networks of this kind.

By a single segment is meant a stretch of railway that connects only two towns. The theorem can be proved by assigning to each railway network, in a unique way, a symbol $\{a_1,\ a_2,\ \dots,\ a_{n-2}\}$, whose $n-2$ elements can be selected independently[6] from any of the numbers 1, 2, . . ., n. There are n^{n-2} such symbols, and this fact, together with the one-to-one correspondence between networks and symbols, will complete the proof.

In the case $n = 2$, the empty symbol corresponds to the only possible network, consisting of just one single segment that connects both towns. If $n > 2$, we denote the towns by the numbers 1, 2, . . ., n and specify them in a fixed sequence. The towns at which only one segment terminates we call the endpoints. [Every network has endpoints] for otherwise there would be at least two segments terminating at each town, and there would be at least $\dfrac{2n}{2} = n$ segments.[7]

In order to define the symbol belonging to a given net for $n > 2$, we proceed as follows.

Let b_1 be the first town which is an endpoint of the net, and a_1 the town which is directly joined to b_1. Then a_1 is the first element of the symbol. We now strike out the town b_1 and the segment $b_1\,a_1$. There remains a net containing $n-2$ segments that connects $n-1$ towns in such a way that one can travel from each town to any other.

If $n-1 > 2$ also, then one determines the town a_2 with which the first endpoint b_2 of the new net is directly connected. We take a_2 as the next element of the symbol. Then we strike out the town b_2 and the segment $b_2\,a_2$. We obtain a net with $n-3$ segments and the same properties.

We continue this procedure until we finally obtain a net with only one segment joining 2 towns. Then nothing more is included in the symbol.

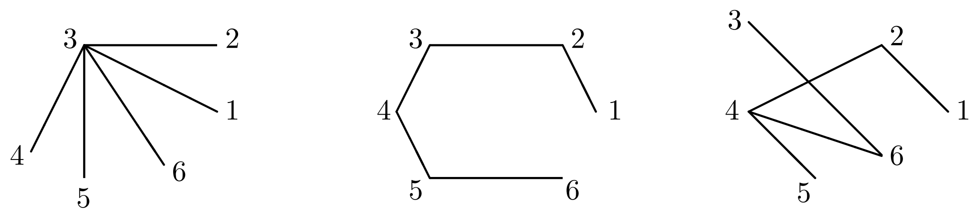
Examples:

Each town at which m segments terminate occurs exactly $m-1$ times in the symbol. For, in the formation of the symbol by successively removing segments, a town appears in the symbol only when one of its incident edges is removed, except in the case that this edge is the last one having that town as endpoint.

[6] The entries (elements) of a symbol may be repeated.

[7] See Exercise (3.7).

Nets:



Symbols: $\{3,\ 3,\ 3,\ 3\}$ $\{2,\ 3,\ 4,\ 5\}$ $\{2,\ 4,\ 6,\ 4\}$

Conversely, if we are given a particular symbol $\{a_1,\ a_2,\ \ldots,\ a_{n-2}\}$, other than the empty symbol, then we write down the numbers 1, 2, …, n, and find the first number that does not appear in the symbol. Let this be b_1. Then we connect the towns b_1 and a_1 by a segment. We now strike out the first element of the symbol and the number b_1.

If $\{a_2,\ a_3,\ \ldots,\ a_{n-2}\}$ is also not the empty symbol, then we find b_2, the first of the $n-1$ remaining numbers that does not appear in the symbol. Connect the towns b_2 and a_2. Then strike out the number b_2 and the element a_2 in the symbol.

In this way we eventually obtain the empty symbol. When that happens, we join the last two towns not yet crossed out.

That the system of segments obtained by this construction actually is a net, and that this net and no other actually gives rise to the given symbol, follows from an induction argument. For, if a net is represented by a symbol, then the towns which do not appear in the symbol are just the endpoints of the net. As the segment $b_1\,a_1$ is the only line ending at b_1, it [segment $b_1\,a_1$] must appear in the net. But we may assume that we have proved that the symbol $\{a_2,\ a_3,\ \ldots, a_{n-2}\}$ corresponds to just one net connecting all the towns except b_1, and that this net was obtained by the construction, so that the truth of the proposition follows for the symbol $\{a_1,\ a_2,\ \ldots,\ a_{n-2}\}$.

In modern language, Prüfer has proven that the number of distinct trees on n fixed vertices is n^{n-2}, $n \geq 2$. To compare with Cayley's formula, see Exercise (3.1). Prüfer begins by assigning to each tree with n vertices a "symbol" consisting of $n-2$ numbers (or characters) taken from the labels of the vertices. Moreover, he establishes that each tree corresponds to only one symbol, and each symbol corresponds to only one tree. Thus, the problem of counting trees is reduced to the problem of counting sequences of length $n-2$ taken from a set of n numbers (or characters), where the characters may be repeated. Two symbols are considered the same if and only if all corresponding entries are the same. Counting symbols is then much easier than counting trees (Exercise 3.9). Before a detailed study of Prüfer's paper, the reader may wish to compute symbols for a few trees (see Exercise (3.2)) and identify the trees associated to a few symbols (see Exercises (3.3), (3.16)).

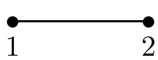
Prüfer uses the notation $\{a_1,\ a_2,\ \ldots\ ,\ a_{n-2}\}$ to denote his symbol, which should not be confused with modern set notation. Today such symbols representing sequences of characters might instead be written as $(a_1,\ a_2,\ \ldots\ ,\ a_{n-2})$. To avoid confusion, when not quoting the original paper, we write a Prüfer symbol as $a_1, a_2, \ldots, a_{n-2}$, without delimiters. Building on Prüfer's own words, let's develop a recursive construction for these symbols. Given a tree T with n vertices, let $\mathcal{S}(T)$ denote the symbol corresponding to T. If $n = 2$, then $\mathcal{S}(T)$ is the empty symbol __ (no entries in the symbol). When $n > 2$, how is the first entry in $\mathcal{S}(T)$ constructed? Prüfer writes "Let b_1 be the first town which is an endpoint of the net, and a_1 the town which is directly joined to b_1. Then a_1 is the first element of the symbol." Is it clear how a_1 is constructed? Could a_1 possibly have two different values, depending on what town(s) b_1 is connected to? (See Exercise 3.10.) Next: "We now strike out the town b_1 and the segment b_1a_1." Is the graph that remains still a tree (Exercise 3.11)? Prüfer does not state a modern recursive definition for $\mathcal{S}(T)$ (Exercise 3.12), but instead explicitly defines a_2, the second entry in the symbol, suggesting an iterative construction in which a_3, a_4, …, a_{n-2} would be defined in similar fashion.

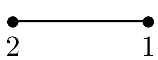
In this way each tree T corresponds to some symbol $\mathcal{S}(T)$. Does every symbol, however, correspond to one tree? Prüfer writes: "Conversely, if we are given a particular symbol $\{a_1, a_2, \ldots, a_{n-2}\}$, other than the empty symbol, then we write down the numbers 1, 2, …, n, and find the first number that does not appear in the symbol. Let this be b_1. Then we connect the towns b_1 and a_1 by a segment." Can this description be used to define an algorithm for constructing trees from symbols? (See Exercise 3.15). For a symbol $\sigma = a_1, a_2, \ldots, a_{n-2}$, let $\mathcal{T}(\sigma)$ denote Prüfer's construction of a network from σ. So far, for $n > 2$, $\mathcal{T}(\sigma)$ contains the vertices b_1 and a_1 together with the edge connecting b_1 to a_1. Prüfer continues: "We now strike out the first element of the symbol and the number b_1." Can the reader envision how this description might lead to a recursive construction of $\mathcal{T}(\sigma)$? (See Exercise 3.17).

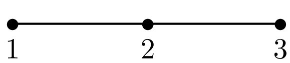
Prüfer's exposition, however, suggests an iterative construction of $\mathcal{T}(\sigma)$, since a formulaic description of b_2 is given without stating that the process could be repeated for the symbol $a_2, a_3, \ldots, a_{n-2}$. He writes: "If $\{a_2, a_3, \ldots, a_{n-2}\}$ is also not the empty symbol, then we find b_2, the first of the $n-1$ remaining numbers that does not appear in the symbol. Connect the towns b_2 and a_2." (See Exercise 3.18.) Prüfer maintains that there is a one-to-one correspondence between networks and symbols,[8] and this should follow from an induction argument (Exercise 3.19).

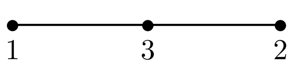
Exercise 3.1. Let's compare the statement of Cayley's result to that of Prüfer's. Cayley writes: "The number of trees which can be formed with $n+1$ given knots … is $(n+1)^{n-1}$," while Prüfer concludes that the number of networks (labeled trees) connecting n-many towns is n^{n-2}. Does n signify the same quantity for both authors? How can these results be reconciled?

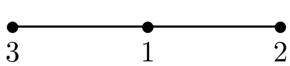
Exercise 3.2. Find the Prüfer symbols of the following labeled trees. Explain your solutions.

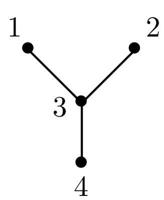
(a) 

(b) 

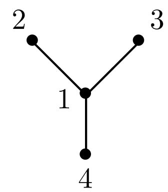
(c) 

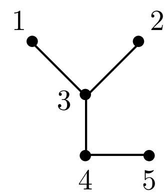
(d) 

(e) 

(f) 

[8] "[Eine] eineindeutige Zuordnung der Netze und Symbole" in the original German [20].

(g) 

(h) 

Exercise 3.3. Find the labeled trees which have the following Prüfer symbols σ. Note that the symbols are written without brackets "$\{\dots\}$" or parentheses "$(\dots)$." Be sure to explain your solutions.

(a) The symbol $\sigma = 1$, where the labels of the vertices are 1, 2, 3. In Prüfer's notation, $a_1 = 1$ is the (first) element of this symbol. What is the value of b_1, the first vertex that does not appear in the symbol? To what vertex is b_1 connected? When b_1 is deleted from the vertices 1, 2, 3, how should the remaining two vertices be treated in Prüfer's algorithm to construct a tree from σ?

(b) The symbol $\sigma = 1,\ 1$, where the labels of the vertices are 1, 2, 3, 4. In Prüfer's notation, $a_1 = 1$ and $a_2 = 1$ are the elements of the symbol. What are the values of b_1 and b_2? To what vertices are b_1 and b_2 connected? When b_1 and b_2 are deleted from the vertices 1, 2, 3, 4, how should the remaining two vertices be treated in the algorithm to construct a tree from σ?

(c) The symbol $\sigma = 1,\ 1,\ 4$, where the labels of the vertices are 1, 2, 3, 4, 5.

(d) The symbol $\sigma = 1,\ 1,\ 2$, where the labels of the vertices are 1, 2, 3, 4, 5.

Exercise 3.4. Let G be a connected graph. We say that G is minimally connected if the removal of any edge of G (without deleting any vertices) results in a disconnected graph.

(a) Show that a connected, minimally connected graph has no cycles.

(b) Show that a connected graph with no cycles is minimally connected.

(c) Why is a connected, minimally connected graph equivalent to a graph being a tree?

Exercise 3.5. Prüfer uses the term "endpoint" to designate a town at which only one railway segment terminates. An "endpoint" is recognized today as a vertex of degree one, and is often called a *leaf* in modern terminology. Let's carefully examine why every tree must have at least one leaf. Suppose that T is a tree on n vertices and every vertex v of T has degree two or greater. Conclude that T must contain a closed cycle.

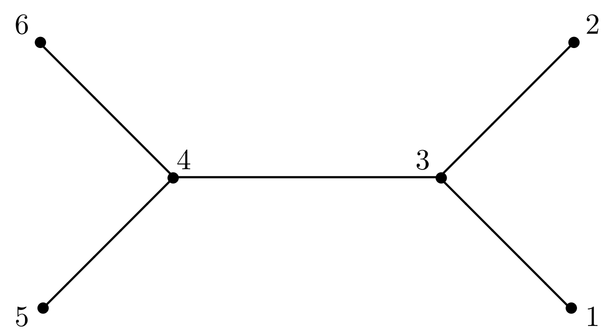
Exercise 3.6. The goal of this exercise is to prove, via induction, that every tree on n vertices has exactly $n-1$ edges, no matter how the tree is configured. Explain why the result holds for $n=2$. As an inductive hypothesis, suppose that if T is a tree on n vertices, then T has $n-1$ edges. Let S be a tree on $n+1$ vertices. Delete a leaf (vertex of degree one) from S and the edge connected to the leaf. Is the graph formed by these deletions still a tree? Why? Now, carefully apply the inductive hypothesis to finish the argument.

Exercise 3.7. Prüfer argues that every network (tree) must have at least one endpoint (leaf) by using a proof by contradiction. Suppose that some tree on n vertices has no leaves. Then every vertex must have degree two or greater. The degree sum of all vertices must then be at least $2n$. Since each edge is counted twice in the degree sum, there must be at least $\frac{2n}{2}=n$ edges in the tree. This contradicts that a tree on n vertices has exactly $n-1$ edges. What must be known apriori to make this argument valid? Hint: examine the proof in Exercise 3.6 that every tree on n vertices has exactly $n-1$ edges.

Exercise 3.8. Prove that every tree (with more than one vertex) contains at least two leaves.

Exercise 3.9. Using modern notation, let $V=\{v_1,\ v_2,\ \ldots\ ,\ v_n\}$ be a set of vertices. How many sequences of length $n-2$ are there using the characters $v_1,\ v_2,\ \ldots,\ v_n$, where characters may be repeated? Two sequences $(\alpha_1,\ \alpha_2,\ \ldots\ ,\ \alpha_{n-2})$ and $(\beta_1,\ \beta_2,\ \ldots\ ,\ \beta_{n-2})$ are counted as the same if and only if $\alpha_1=\beta_1,\ \alpha_2=\beta_2,\ \ldots,\ \alpha_{n-2}=\beta_{n-2}$. Be sure to explain your answer.

Exercise 3.10. Suppose that a network (tree) has several endpoints (leaves). How is the first endpoint (leaf) chosen? Is every endpoint connected to exactly one town (vertex)? Why? Find the first entry of the Prüfer symbol of the following tree, and be sure to explain your answer.



Exercise 3.11. Let T be a tree on n ordered vertices and let b_1 be a leaf of T (the first leaf, if necessary). Let a_1 be the vertex to which b_1 is connected by an edge b_1a_1, and let T' be the graph constructed from T by deleting the vertex b_1 and deleting the edge b_1a_1 (do not delete the vertex a_1). Prove that T' is a tree, by showing that either

(a) T' is a connected graph that contains no cycles; or

(b) T' is a connected, minimally connected graph.

Which argument, (a) or (b), does Prüfer's paper suggest? How many vertices does T' contain?

Exercise 3.12. Let T be a tree on n ordered vertices and let T' be the tree on $n-1$ vertices constructed in Exercise 3.11. If $n > 3$, define $\mathcal{S}$ recursively by

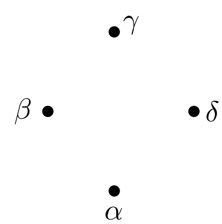
$$\mathcal{S}(T) = a_1, \mathcal{S}(T'),$$

i.e., $\mathcal{S}(T) = (a_1,\ \mathcal{S}(T'))$, to use modern delimiters. Here, a_1 is given in Exercise 3.11. Does this construction match Prüfer's description "If $n-1 > 2$ also, then one determines the town a_2 with which the first endpoint b_2 of the new net is directly connected. We take a_2 as the next element of the symbol. Then we strike out the town b_2 and the segment $b_2\,a_2$. We obtain a net with $n-3$ segments and the same properties. We continue this procedure until we finally obtain a net with only one segment joining 2 towns. Then nothing more is included in the symbol."

Which construction do you find easier to understand, the recursive definition or Prüfer's description? Which would be easier to implement? Why?

Exercise 3.13. Apply the recursive definition of $\mathcal{S}(T)$ in Exercise 3.12 to compute the Prüfer symbol of the tree appearing at the end of Exercise 3.10.

Exercise 3.14. Find all 16 trees on four fixed vertices α, β, γ and δ, arranged as follows (without edges drawn).



Using the ordering $\alpha < \beta < \gamma < \delta$, ($\alpha = 1$, $\beta = 2$, $\gamma = 3$, $\delta = 4$), find the Prüfer symbol of each of these 16 trees, and compare your solution to Exercise 2.4. Is there a systematic pattern to the construction of the Prüfer symbols?

Exercise 3.15. Given a symbol $\sigma = a_1, a_2, \ldots, a_{n-2}$, let's examine Prüfer's construction of a graph from σ. First he supposes that the vertices are given as the set of numbers $\{1,\ 2,\ 3,\ \ldots,\ n\}$, although any fixed set of n-many characters $V = \{v_1,\ v_2,\ v_3,\ \ldots,\ v_n\}$ could be used, provided that the elements of V are ordered in some way, e.g.,

$$v_1 < v_2 < v_3 < \ \ldots\ < v_n.$$

Thus, we can "find the first number that does not appear in the symbol." Let's use $V = \{1,\ 2,\ 3,\ \ldots,\ n\}$ as the vertex set. Then $\{a_1,\ a_2,\ \ldots,\ a_{n-2}\}$ is a subset of $\{1,\ 2,\ 3,\ \ldots,\ n\}$. Using set-theoretic notation, this "first number" would be given via the construction

$$b_1 = \min. S, \quad S = \{1,\ 2,\ 3,\ \ldots,\ n\} - \{a_1,\ a_2,\ \ldots,\ a_{n-2}\},$$

where the minus sign indicates that the elements of the set $\{a_1,\ a_2,\ \ldots,\ a_{n-2}\}$ are deleted from $\{1,\ 2,\ 3,\ \ldots,\ n\}$, and "min." denotes the minimum of the resulting set S.

(a) Explain why $\{1,\ 2,\ 3,\ \ldots,\ n\} - \{a_1,\ a_2,\ \ldots,\ a_{n-2}\}$ is a finite, non-empty set, where $n > 2$.

(b) For $n > 2$, does $\{1,\ 2,\ 3,\ \ldots,\ n\} - \{a_1,\ a_2,\ \ldots,\ a_{n-2}\}$ always have a least element? Why or why not?

(c) For $n = 2$, σ is the empty symbol. Draw a tree on two vertices that corresponds to the empty symbol, and label the vertices using $V = \{v_1,\ v_2\}$. Did you draw the tree as a horizontal line? If so, did you label the right-hand or the left-hand vertex as v_1? Do both of these trees count as the same tree from Prüfer's (and Cayley's) point of view? Did you draw the tree as a vertical line? If so, did you label the upper or lower vertex as v_1? Do both of these trees count as the same from Prüfer's (and Cayley's) point of view? Explain why any one-segment tree can be rotated or stretched to achieve a tree that represents one and the same tree for the purpose of counting.

(d) For $n > 2$, draw a segment (edge) that connects the vertices labeled b_1 and a_1. Does the relative position of b_1 and a_1 matter for the purpose of counting trees?

(e) Explain why $a_1 \neq b_1$ from the construction of b_1.

Exercise 3.16. Construct the tree with Prüfer symbol $\sigma = 5,\ 7,\ 5,\ 5,\ 4,\ 5$ using the vertex set 1, 2, 3, 4, 5, 6, 7, 8. In Prüfer's notation, $a_1 = 5$, $a_2 = 7$, $a_3 = 5$, $a_4 = 5$, $a_5 = 4$, $a_6 = 5$. What are the values of b_1, b_2, b_3, b_4, b_5 and b_6? To what vertices are the b_is connected? Finally, when all the b_is are deleted from the vertices 1, 2, 3, 4, 5, 6, 7, 8, how should the remaining two vertices be treated in Prüfer's algorithm to construct a tree from σ? Justify your answer by citing a relevant passage from Prüfer's paper.

Exercise 3.17. Given a symbol $\sigma = a_1, a_2, \ldots, a_{n-2}$, where each a_i, $1 \le i \le n-2$, is an element of the vertex set $V = \{1,\ 2,\ 3,\ \ldots,\ n\}$, then by striking out "the first element of the symbol," we obtain a new symbol $\sigma' = a_2, a_3, \ldots, a_{n-2}$, where each a_i, $2 \le i \le n-2$, is now in the vertex set

$$V' = \{1,\ 2,\ 3,\ \ldots,\ n\} - \{b_1\},$$

constructed by striking out "the number b_1" from the old vertex set.

(a) Is $a_1 \in V'$? Justify your answer.

(b) When $n = 2$ and σ is the empty symbol, what tree is $\mathcal{T}(\sigma)$?

(c) When $n > 2$, suppose by induction that $\mathcal{T}(\sigma')$ is a tree on the vertex set V'. Must $\mathcal{T}(\sigma')$ have a vertex labeled a_1?

(d) Let $(b_1 a_1)$ be the tree with two vertices b_1, a_1, and one edge connecting b_1 to a_1. Define a new graph

$$(b_1 a_1) \vee \mathcal{T}(\sigma'),$$

called the graft of $(b_1 a_1)$ with $\mathcal{T}(\sigma')$, constructed by identifying the vertices labeled a_1 in both factors, i.e., the vertex labeled a_1 in $(b_1 a_1)$ is placed on the vertex labeled a_1 in $\mathcal{T}(\sigma')$, and the one edge connecting b_1 to a_1 is added to the tree $\mathcal{T}(\sigma')$. Carefully explain why $(b_1 a_1) \vee \mathcal{T}(\sigma')$ is a tree, using the inductive hypothesis in (c).

(e) For $n > 2$, define $\mathcal{T}(\sigma)$ recursively by

$$\mathcal{T}(\sigma) = (b_1 a_1) \vee \mathcal{T}(\sigma').$$

When $n = 3$ and $V = \{1,\ 2,\ 3\}$, construct the three separate trees $\mathcal{T}(1)$, $\mathcal{T}(2)$, $\mathcal{T}(3)$ using the recursive definition. Compare your solution to the trees in Figure 2.1 using $\alpha = 1$, $\beta = 2$, and $\gamma = 3$.

(f) For $n = 8$, construct the tree $\mathcal{T}(5, 7, 5, 5, 4, 5)$ using the vertex set 1, 2, 3, 4, 5, 6, 7, 8. Compare your solution to Exercise (3.16). Be sure to explain your work.

Exercise 3.18. From Prüfer's paper, an iterative formula for b_2 would be

$$b_2 = \min. S, \quad S = \big(\{1,\ 2,\ 3,\ \dots,\ n\} - \{b_1\}\big) - \{a_2,\ a_3,\ \dots,\ a_{n-2}\}.$$

(a) Show that the above formula for b_2 agrees with the resulting value of b_2 from the recursive construction

$$\mathcal{T}(\sigma) = (b_1 a_1) \vee \mathcal{T}(\sigma').$$

(b) If $a_3, a_4, \dots, a_{n-2}$ is not the empty symbol, find an iterative formula for b_3, similar to b_2 above.

(c) Given the Prüfer symbol $\sigma = a_1,\ a_2,\ \dots,\ a_{n-2}$, explain why $b_1, b_2, b_3, \dots, b_{n-2}$ are all distinct elements of $\{1,\ 2,\ 3,\ \dots,\ n\}$. How are the remaining two elements of

$$\{1,\ 2,\ 3,\ \dots,\ n\} - \{b_1,\ b_2,\ b_3,\ \dots,\ b_{n-2}\}$$

treated in Prüfer's iterative construction of a tree from σ?

(d) Which method, the iterative or the recursive construction, is easier to understand? easier to implement? Why do you think so?

Exercise 3.19. Let $\mathcal{S}_n$ denote the function that assigns a symbol to a given tree with n ordered vertices. Conversely, let $\mathcal{T}_n$ denote the function that assigns a tree to a given symbol $\sigma = a_1, a_2, \dots, a_{n-2}$, where each a_i is an element of a specified vertex set V containing n ordered elements. For simplicity, you may assume that V consists of n distinct town names, ordered alphabetically. Set

$$\mathcal{S}_n \circ \mathcal{T}_n(\sigma) = \mathcal{S}_n(\mathcal{T}_n(\sigma)), \quad \mathcal{T}_n \circ \mathcal{S}_n(T) = \mathcal{T}_n(\mathcal{S}_n(T)).$$

(a) If T is a tree on two vertices, explain why $\mathcal{T}_2 \circ \mathcal{S}_2(T) = T$. If σ is the empty symbol, show that $\mathcal{S}_2 \circ \mathcal{T}_2(\sigma) = \sigma$.

(b) Prove by induction on n that $\mathcal{T}_n \circ \mathcal{S}_n(T) = T$ and $\mathcal{S}_n \circ \mathcal{T}_n(\sigma) = \sigma$, where $n > 2$.

(c) Explain why there is a one-to-one correspondence between trees on n fixed vertices and symbols of length $n - 2$ chosen from a vertex set V of n ordered elements.

4 Borůvka's Solution to a Minimization Problem

Otakar Borůvka (1899–1995) was born in Uherský Ostroh, a town in the region of Moravia, formerly belonging to Austria-Hungary, now part of the Czech Republic. In 1926 he published two papers [2, 3] that would later lead to some of the most efficient solutions to what today are called combinatorial optimization problems [12]. The original problem that Borůvka sought to solve can be easily stated, and is of practical value. Given n towns in some region, how should an electrical power network be constructed so that:

1. every town is connected to the network; and
2. the total length of the network is the shortest possible.

This has become known as the minimum spanning tree problem, and has been a topic of research in computer science and algorithm design. Textbooks often cite the work of Kruskal [16] and Prim [19] from the late 1950s for a solution to the minimum spanning tree problem, although both of these authors acknowledge the work of Borůvka in their own papers. In his 1926 publications, Borůvka uses none of the terminology from modern graph theory, not even the word "tree." Consequently, these papers can be read without any specialized knowledge of computer science.

The young Otakar studied mathematics at the Czech Technical University and Masaryk University, both in Brno. He worked closely with the renowned Matyáš Lerch and Eduard Čech (one of the founders of topology and differential geometry). Čech directed Borůvka's interest to geometry and arranged his stay with Elie Cartan in Paris during the years 1926–1927, where he lectured about his pioneering 1926 papers. In 1934 he became a Professor at Masaryk University, and in 1953 a corresponding member of the Czechoslovak Academy (ordinary member 1965). In 1959 he received the State Prize of Czechoslovakia, and in 1965 he founded the *Journal Archivum Mathematicum.* Borůvka's interests in mathematics were broad, and he authored the influential textbooks *Grundlagen der Gruppoid und Gruppentheorie* [5] and *Lineare Differentialtransformationen 2. Ordnung* [6], both translated and published in English as *Foundations of the Theory of Groupoids and Groups* [8] and *Linear Differential Transformations of the Second Order* [7] respectively.

Otakar Borůvka's early pioneering work offered a solution to the problem of finding the most efficient method of connecting certain towns with an electrical network. The problem was originally communicated to him by a friend, Jindřich Saxel, an employee of West-Moravian Powerplants, and concerned providing electrical power to the South Moravian Region (presently part of the Czech Republic) [18]. Certainly every town in this region should be connected to the electrical grid, and, moreover, the towns should be connected so that "the total length of the net would be the shortest possible" [3, 18]. Today this problem can be cast in terms of graph theory. An electrical network forms a graph with vertices given by the towns and edges given by electrical cables that directly connect two towns. Of all possible electrical networks, which one(s) is (are) the least expensive to construct? Let's suppose that the cost of construction is directly proportional to the total length of the needed cable. Borůvka writes [3, 18]:

> There are n points in the plane (in space) whose mutual distances are all different. We wish to join them by a net such that:
>
> 1. Any two points are joined either directly or by means of some other points.
> 2. The total length of the net would be the shortest possible.

Is a network satisfying properties (1) and (2) above necessarily a tree (Exercise 4.1)? If so, of all the possible n^{n-2} trees on n fixed vertices, how would the tree(s) of minimum total length be found? For n as small as 10, there would be a total of $10^8 = 100,000,000$ trees to consider. Borůvka proposes a simple algorithm to find such a net of minimum total length, based on the guiding principle "I shall join each of the given points with the point nearest to it" [3, 12]. Of course, given points v_1, v_2, v_3, ... in the plane, if the closest point to v_1 is v_2, then it is not necessarily the case that the closest point to v_2 is v_1. For example, consider the points with xy-coordinates given by

$$v_1(1,\ 0), \quad v_2(3, 0), \quad v_3(4,\ 0).$$

Then the closest point to v_1 is v_2, while the closest point to v_2 is v_3. On the other hand, given n points in the plane, whose mutual distances are all different, would a connected graph result if the only connections made are those resulting from connecting a vertex to its nearest neighbor? (See Exercise 4.2.)

Let's read Borůvka's solution to finding a connected network of minimum total length. In 1926 he published two papers on this subject. The first "On a Certain Minimal Problem," [2, 12, 18] is a rather algebraic account of the problem, while the second "A Contribution to the Solution of a Problem on the Economical Construction of Power Networks" [3, 12, 18] is a verbal discussion of the solution to a particular example. The second paper greatly illuminated the algebraic language of the first.

A Contribution to the Solution of a Problem on the Economical Construction of Power Networks

Dr. Otakar Borůvka

In my paper "On a Certain Minimal Problem," I proved a general theorem, which, as a special case solves the following problem:

There are n points in the plane (in space) whose mutual distances are all different. We wish to join them by a net such that:

1. *Any two points are joined either directly or by means of some other points.*
2. *The total length of the net would be the shortest possible.*

It is evident that a solution of this problem could have some importance in electrical power network designs; hence I present the solution briefly using an example. . . . .

I shall give the solution of the problem in the case of 40 points[9] given in Fig. 1.

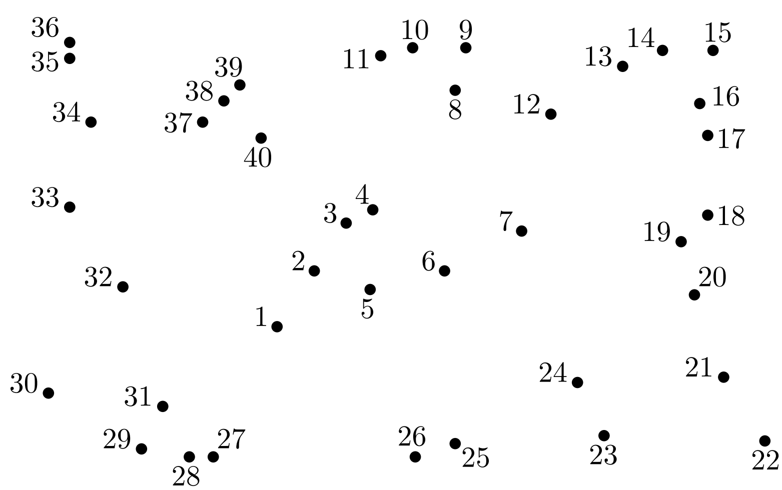


Fig. 1.

I shall join each of the given points with the point nearest to it. Thus, for example, point 1 with point 2, point 2 with point 3, point 3 with point 4 (point 4 with point 3), point 5 with point 2, point 6 with point 5, point 7 with point 6, point 8 with point 9 (point 9 with point 8), etc. I shall obtain a sequence of polygonal strokes 1, 2, . . . , 13 (Fig. 2).

[9] Borůvka only labeled the points 1 through 9 in his original paper. We have included labels of all points for later reference.

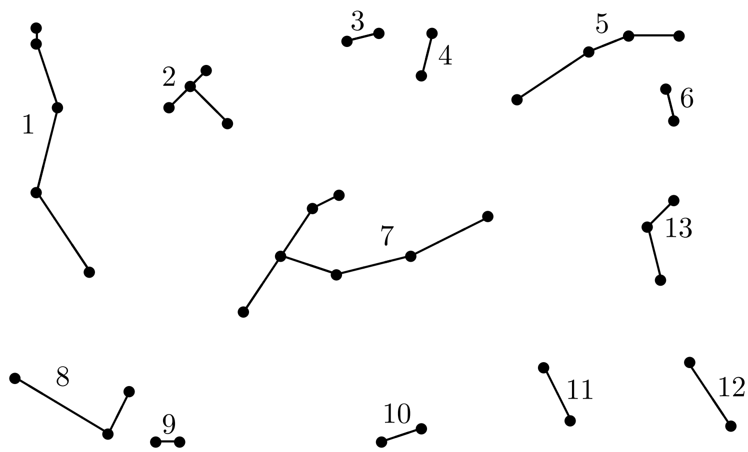


Fig. 2.

I shall join each of these strokes with the nearest stroke in the shortest possible way. Thus, for example, stroke 1 with stroke 2 (stroke 2 with stroke 1), stroke 3 with stroke 4 (stroke 4 with stroke 3), etc. I shall obtain a sequence of polygonal strokes 1, 2, 3, 4 (Fig.3).

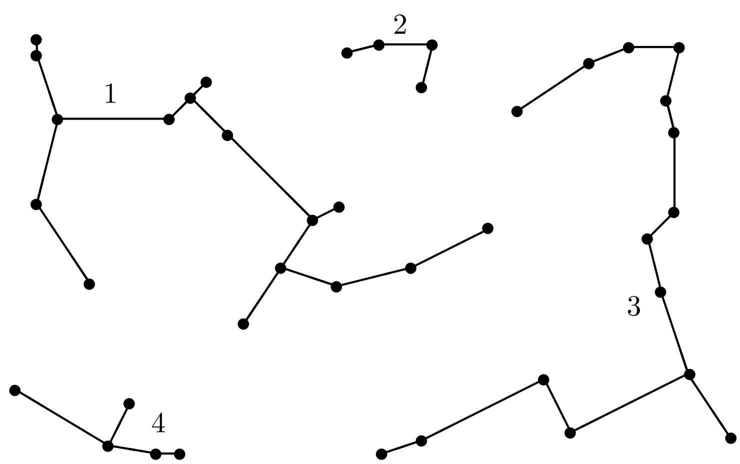


Fig. 3.

I shall join each of these strokes in the shortest way with the nearest stroke. Thus stroke 1 with stoke 3, stroke 2 with stroke 3 (stroke 3 with stroke 1), stroke 4 with stroke 1. I shall finally obtain a single polygonal stroke (Fig. 4)[10] which solves the given problem.

In moving from Figure 1 to Figure 2, how can we decide which points (vertices) to connect? The verbal statement "I shall joint each of the given points with the point nearest to it" provides an excellent intuitive answer to this question. Let's compare this to the algebraic constructions of

[10] In the original paper [3], Figure 4 is rotated 180°.

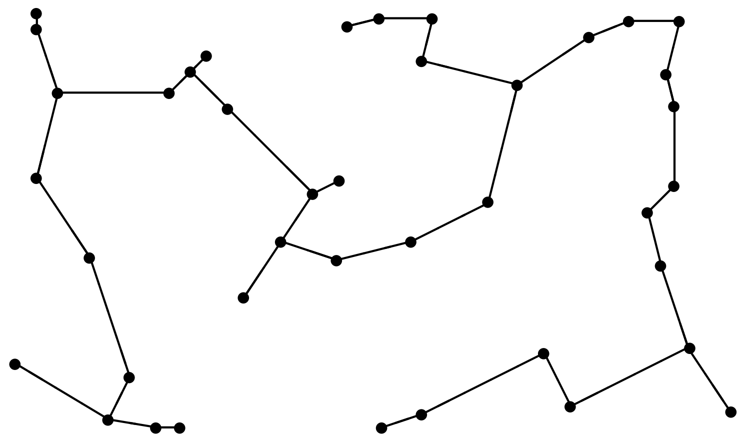


Fig. 4.

Borůvka's first paper "On a Certain Minimal Problem" [2], authored without any modern terms from graph theory.

ON A CERTAIN MINIMAL PROBLEM

Otakar Borůvka

In this article I am presenting a solution of the following problem:

Given a matrix M of numbers $r_{\alpha\beta}$ $(\alpha,\ \beta = 1,\ 2,\ \dots,\ n;\ n \geq 2)$, all positive and pairwise different, with the exception of $r_{\alpha\alpha} = 0$ and $r_{\alpha\beta} = r_{\beta\alpha}$.

From that matrix a set of nonzero and pairwise different numbers should be chosen such that

(1) For any p_1, p_2, mutually different natural numbers $\leq n$, it would be possible to choose a subset of the form

$$r_{p_1 c_2},\ r_{c_2 c_3},\ r_{c_3 c_4},\ \dots,\ r_{c_{q-2} c_{q-1}},\ r_{c_{q-1} p_2}.$$

(2) The sum of its elements would be smaller than the sum of elements of any other subset of nonzero and pairwise different numbers, satisfying the condition (1)[11]

Solution. Let f_0 be an arbitrary choice of the numbers α and let $[f_0\, f_1]$ be the smallest of the numbers $[f_0\, \gamma_0]$, $\gamma_0 \neq f_0$. ...

For the example in Figure 1 of Borůvka's paper, the matrix M would have entries $r_{\alpha\beta}$, $\alpha = 1, 2, 3, \dots, 40$, and $\beta = 1, 2, 3, \dots, 40$, where $r_{\alpha\beta}$ denotes the actual distance between point α and point β. With the vertices (1–40) labeled across the rows and columns, the matrix would begin as (hypothetical distances, given in kilometers):

[11] For the sake of brevity I shall use the symbol $[\alpha\, \beta]$ instead of $r_{\alpha\beta}$ from now on. (The footnote actually appears in the original paper.)

M	1	2	3	4	...	40
1	0	8	15	19.6	...	29
2	8	0	7	10	...	18
3	15	7	0	4	...	15.1
4	19.6	10	4	0	...	12
⋮	⋮	⋮	⋮	⋮		⋮
40	29	18	15.1	12	...	0

Thus, $r_{1,2} = 8$ and $r_{4,40} = 12$. For further work with M, see Exercises 4.3 and 4.4.

The first step in Borůvka's algorithm can be summarized as follows. Let $V = \{v_1, v_2, v_3, \ldots, v_n\}$ be a set of n-many vertices (in the xy-plane) with mutually different distances. Let $[v_i\ v_j]$ denote the distance between v_i and v_j. Then connect each vertex v_i to some vertex v_m, where

$$[v_i\ v_m] = \min_{j \neq i} [v_i\ v_j].$$

Following Borůvka, let $\mathfrak{G}$ denote the resulting graph. If $\mathfrak{G}$ is connected, stop. If not, $\mathfrak{G}$ can be expressed as the union of a finite number of connected pieces (connected components) as in Figure 2, where the components are called "polygonal strokes" and numbered 1, 2, 3, ..., 13. In Borůvka's papers the idea of a connected component was not expressed as an independent concept, and he outlines a very detailed construction in [2] to arrive at a sequence of subgraphs G_0, G_1, G_2, ..., $G_{\ell-1}$ that plays the role of the polygonal strokes in Figure 2. Of course, how should the G_i's be connected among themselves? In the example of 40 points, when moving from Figure 2 to Figure 3, certain of the G_i's are connected and in very specific ways. From [2], we read:

> Let us put $H_\lambda \equiv G_\lambda$ $(\lambda = 0, 1, \ldots, \ell - 1)$. The sequence of sets $\mathfrak{G}$ contains either just the set G [the set G_0] or more sets. In the first case, let us put
>
> $$J \equiv \mathfrak{G},$$
>
> in the second case, let κ_λ be any of the indices [points] which occur in the elements of the set H_λ; α_1, β_1 be two of the number λ; $[k_{\alpha_1\beta_1}\ k_{\beta_1\alpha_1}]$ be the smallest of the numbers $[\kappa_{\alpha_1}\ \kappa_{\beta_1}]$ when $\alpha_1 \neq \beta_1$, $[k_{\alpha_1\beta_1}\ k_{\beta_1\alpha_1}] = 0$ when $\alpha_1 = \beta_1$; M_1 the matrix of numbers $[k_{\alpha_1\beta_1}\ k_{\beta_1\alpha_1}]$ $(\alpha_1, \beta_1 = 0, 1, 2, \ldots, \ell - 1)$[12]. ...

Let's examine how this new matrix M_1 is constructed. From the example of 40 points, consider the polygonal stroke (connected component) corresponding to $\lambda = 11$ in Figure 2. There are two indices (points) in this component, namely points 23 and 24 (labeled in Figure 1). Now consider the components corresponding to $\alpha_1 = 11$ and $\beta_1 = 12$. The 12th component contains the points 21 and 22. In Borůvka's notation, $[k_{11,12}\ k_{12,11}]$ denotes the smallest of four numbers given by the distances from any point in component 11 to any point in component 12. This is the smallest number in the following submatrix of M (point labels across the rows and columns):

	23	24
21	16	17.9
22	20.1	24.5

[12] The matrix M_1 is obviously symmetrical ... and its order equals at most the largest integer $\leq \frac{n}{2}$. This footnote appears in the original paper and provides a key insight into the running time of Borůvka's algorithm (Exercise 4.15).

Thus, $[k_{11,12}\ k_{12,11}] = [21\ \ 23] = 16$ and this minimum distance is achieved by connecting points 21 and 23. For more practice computing the values $[k_{\alpha_1\beta_1}\ k_{\beta_1\alpha_1}]$ see Exercise 4.5.

In Borůvka's example, the matrix M_1 is given by the smallest distance between component α and component β for $\alpha = 1, 2, 3, \ldots, 13$ and $\beta = 1, 2, 3, \ldots\ 13$. Writing the component labels across the rows and columns, we have:

$\mathbf{M_1}$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
1	0	8.5	36	45	52	75.5	20	15.5	23	41.5	57.5	75	70
2	8.5	0	17.9	24.5	31	55	14.9	35.5	40	44.5	50	65	54
3	36	17.9	0	6.5	19	31.5	19.2	51.5	54	49	41.5	51	41
4	45	24.5	6.5	0	12.1	30.5	18.1	53	54.5	44.2	39.8	49.5	34
5	52	31	19	12.1	0	6.8	15.2	60	59.5	43	34.1	40.9	20.5
6	75.5	55	31.5	30.5	6.8	0	21	75.2	73	50.1	34.5	30	9.9
7	20	14.9	19.2	18.1	15.2	21	0	17	18.2	21.4	20.3	31.1	19.7
8	15.5	35.5	51.5	53	60	75.2	17	0	5.8	32.4	51.7	57	62
9	23	40	54	54.5	59.5	73	18.2	5.8	0	25.1	41.2	64	62.8
10	41.5	44.5	49	44.2	43	50.1	21.4	32.4	25.1	0	17.1	34.2	34.4
11	57.5	50	41.5	39.8	34.1	34.5	20.3	51.7	41.2	17.1	0	16	18.1
12	75	65	51	49.5	40.9	30	31.1	57	64	34.2	16	0	10.9
13	70	54	41	34	20.5	9.9	19.7	62	62.8	34.4	18.1	10.9	0

Borůvka continues [2]: "[Let] $\mathfrak{G}_1 = G^{(1)}, G_1^{(1)}, \ldots, G_{\ell_1-1}^{(1)}$ be the sequence of sets we get from the matrix M_1 in the same way as we got the sequence of sets $\mathfrak{G}$ from the matrix M." Can the reader now connect certain of the polygonal strokes in Figure 2 following this rule (Exercise 4.6)? Is the resulting graph $\mathfrak{G}_1$ connected? If not, can the reader identify a recursive (or iterative) algorithm to produce a connected graph following Borůvka's description (Exercise 4.7)? Borůvka realizes that, in general, the graph $\mathfrak{G}_1$ may not be connected, and he iterates the $\mathfrak{G}$ construction to form $\mathfrak{G}_2$ by use of a matrix M_2, derived from M_1 (Exercise 4.7), again connecting a polygonal stroke to its nearest neighbor (nearest polygonal stroke). This process is iterated until a sequence of graphs is formed

$$J = \mathfrak{G}_0,\ \mathfrak{G}_1,\ \mathfrak{G}_2,\ \mathfrak{G}_3,\ \ldots\ \mathfrak{G}_{u-1},$$

where $\mathfrak{G}_{u-1}$ is connected, which, as Borůvka writes "is a solution of the given problem" [2].

Let's investigate the properties of the graphs $\mathfrak{G}_m(V)$, where m is a positive integer and $V = \{v_1,\ v_2,\ v_3,\ \ldots,\ v_n\}$ is an initial vertex set of points with mutually different positive distances. Using the notation of Exercise 4.7, let $\mathfrak{G}_0(V)$ be the graph formed by the first iteration of Borůvka's algorithm, where each vertex v_i is connected to its closest neighbor. Let V' be the set of connected components of $\mathfrak{G}_0(V)$. Then let $\mathfrak{G}_1(V) = \mathfrak{G}_0(V')$, where the distance between any two components G_α and G_β is given by the minimum of the distances between any vertex $u \in G_\alpha$ and any vertex $w \in G_\beta$. Let $\mathfrak{G}_m(V) = \mathfrak{G}_{m-1}(V')$, $m \geq 1$. Is each graph $\mathfrak{G}_m(V)$ cycle-free? To help answer this question, first consider a connected component of $\mathfrak{G}_0(V)$. Using the nearest neighbor property for the connectedness of vertices along with a proof by contradiction, it can be shown that $\mathfrak{G}_0(V)$ contains no cycles (Exercise 4.8). By induction on m, $\mathfrak{G}_m(V)$ contains no cycles (Exercise 4.9). Additionally, there must be some positive integer c so that $\mathfrak{G}_c(V)$ is connected, since each iteration of $\mathfrak{G}$ adds one or more edges to the entire graph, and there is an upper bound on the number of edges in a cycle-free graph with n vertices. (See Exercises 4.10, 4.11). Thus, $\mathfrak{G}_c(V)$ is a connected graph with no cycles, and is, therefore, a tree.

Of all possible trees on V, why is $\mathfrak{G}_c(V)$ a tree of minimum total edge length, i.e., a minimum spanning tree? Let's summarize Borůvka's solution. First, from the work of Cayley and Prüfer we know that there are n^{n-2} possible trees on n fixed vertices. Of all these trees, the total edge length could be computed for each possibility, and then a tree of minimum total length, T_0, could be chosen. Thus, as is known to Borůvka, there is some tree that solves the problem. Is $T_0 = \mathfrak{G}_c(V)$? If not, which edges of $\mathfrak{G}_c(V)$ would not be edges of T_0? (See Exercises 4.12, 4.13, 4.14).

Borůvka's algorithm is today recognized as one of the most efficient for finding a minimal spanning tree. In practice, the edge length connecting two vertices can be replaced with the actual cost of constructing a connection (electrical or otherwise) between two vertices, and, in general, is referred to as the "weight" of the edge in modern terminology. With the advent of the electronic digital computer, interest in minimal spanning tree algorithms has increased, and in the late 1950s both Kruskal [16] and Prim [19] published their work on this problem. Textbooks on graph theory often cite these two authors, although both Kruskal and Prim refer to Borůvka in their own work. Kruskal begins his algorithm by first ordering the edges by length (weight), and then proceeds through this ordered list, edge by edge, to determine whether adding a given edge would possibly create a cycle (in a potential minimal spanning tree). Prim takes a different point of view, and divides the set of vertices into two disjoint classes, those already chosen by his algorithm to be in a minimal spanning (sub)tree, P, and those that remain, R. From an arbitrary vertex in P, find the shortest edge between this vertex and the vertices of R. His algorithm checks whether the new edge should be part of the minimal spanning tree and updates the sets P and R. The guiding principles behind Borůvka's intuitive approach, however, remain appealing. An entire subject, combinatorial optimization [17], has arisen to study similar problems.

Exercise 4.1. Let G be a graph with vertices given by n points in the plane whose mutual distances are all different positive numbers. Suppose, as Borůvka does, that:

1. any two points of G are joined either directly or by means of some other points; and
2. the total length of the net is the shortest possible.

Must G be a connected graph? Why or why not? Could G possibly contain a cycle? Why or why not? Must G necessarily be a tree? Justify your answer.

Exercise 4.2. Consider the points in the xy-plane given by

$$v_1(0,\,0),\quad v_2(3,\,0),\quad v_3(0,\,4),\quad v_4(2,\,4),\quad v_5(0,10),\quad v_6(1,\,10).$$

Connect each point to its nearest neighbor by an edge. Is the resulting graph connected? Justify your answer.

Exercise 4.3. In Borůvka's description of the matrix M with entries $r_{\alpha\beta}$, what is the physical significance of the requirements

(a) $r_{\alpha\alpha} = 0$

(b) $r_{\alpha\beta} = r_{\beta\alpha}$?

Exercise 4.4. Given the matrix M in Exercise 4.3 (described at the beginning of Borůvka's paper "On a Certain Minimal Problem"), what is the meaning of the condition "Let f_0 be an arbitrary choice of the numbers α and let $[f_0\, f_1]$ be the smallest of the numbers $[f_0\, \gamma_0]$, $\gamma_0 \neq f_0$" ? Specifically, what does $[f_0\, f_1]$ represent? Be sure to explain your answer.

Exercise 4.5. In Borůvka's example of 40 points, just from Figure 2, compute the following values and explain your answer. You may wish to measure the distances on the paper in centimeters (or use visual inspection), and then use the scale that 1 cm corresponds to 1 km on the ground. Also, be sure to state which pair of vertices yields the particular $[k_{\alpha_1\beta_1}\ k_{\beta_1\alpha_1}]$ value.

(a) $[k_{1,2}\ k_{2,1}]$

(b) $[k_{1,7}\ k_{7,1}]$

(c) $[k_{1,8}\ k_{8,1}]$

(d) $[k_{1,9}\ k_{9,1}]$

(e) $[k_{2,7}\ k_{7,2}]$

Exercise 4.6. Following the principle that each polygonal stroke should be connected to its nearest polygonal stroke, use the values in the matrix M_1 to carefully explain which polygonal strokes in Figure 2 are connected, and exactly which pair of vertices are connected in this process. Let $\mathfrak{G}_1$ denote the resulting graph. Is $\mathfrak{G}_1$ connected? Why or why not?

Exercise 4.7. Given a set of vertices $V = \{v_1,\ v_2,\ v_3,\ \ldots,\ v_n\}$, all of whose mutual distances are different positive numbers, let $\mathfrak{G}_0(V)$ denote the graph formed by connecting each vertex $v_i \in V$ to its nearest neighbor. From $\mathfrak{G}_0(V)$ consider a new set of vertices V' given by the connected components of $\mathfrak{G}_0(V)$. Suppose that the distance between any two components G_α, G_β is given by the minimum of the distances between any vertex $u \in G_\alpha$ and any vertex $w \in G_\beta$. Now apply the $\mathfrak{G}_0$ construction to V' and let

$$\mathfrak{G}_1(V) = \mathfrak{G}_0(V').$$

(a) For Borůvka's example of 40 points (vertices) given in Figure 1, what graph results from the construction $\mathfrak{G}_1(V) = \mathfrak{G}_0(V')$? Carefully explain your answer, perhaps using results from Exercise 4.6.

(b) For the example in Figure 1, what graph results from the construction

$$\mathfrak{G}_2(V) = \mathfrak{G}_1(V') = \mathfrak{G}_0((V')')\ ?$$

Justify your answer, forming a new matrix M_2 giving the distances between the connected components of $\mathfrak{G}_0(V')$. Explain how the matrix M_2 can be constructed from the matrix M_1.

(c) Above, is the graph $\mathfrak{G}_2(V)$ connected? Why or why not?

(d) Find a recursive construction for $\mathfrak{G}_m(V)$, where m is a positive integer.

Exercise 4.8. Let G_α be a connected component of $\mathfrak{G}_0(V)$. Use a proof by contradiction to show that G_α is cycle-free. Hint: assume that φ is a cycle of G_α and consider the longest edge in this cycle.

Exercise 4.9. Prove by induction on m that $\mathfrak{G}_m(V)$ contains no cycles. As an inductive hypothesis, suppose that $\mathfrak{G}_m(V)$ has no cycles. Represent each connected component of $\mathfrak{G}_m(V)$ as a dot and connect these dots $(D_1,\ D_2,\ \ldots,\ D_p)$ via Borůvka's closest neighbor algorithm. Why does the resulting graph on the D_is have no cycles? Now, expand each D_i in terms of its underlying graph. Why is $\mathfrak{G}_{m+1}(V)$ cycle-free?

Exercise 4.10. Let G be a graph on n vertices with no cycles. Show that G has at most $n-1$ edges.

Exercise 4.11. Show that there is some positive integer c so that $\mathfrak{G}_c(V)$ is connected.

Exercise 4.12. Let T_0 be a tree of minimum total edge length on n vertices. Use a proof by contradiction to show that T_0 must contain the edges of $\mathfrak{G}_0(V)$. Hint: Let e_0 be an edge of $\mathfrak{G}_0(V)$ connecting vertices u and w. Then either w is the closest neighbor to u or vice versa (u is the closest neighbor to w). Consider the case where w is the closest neighbor to u. Assume that T_0 does not contain the edge e_0. Since T_0 is connected, T_0 must contain a path φ from u to w, which by assumption does not traverse e_0. Let u, v_1, v_2, $\ldots$ v_q w be the sequence of vertices of φ from u to w.

(a) Why is $[u\ v_1] > [u\ w]$?

Let e_1 be the edge of T_0 connecting u to v_1. Form a new graph from T_0 by deleting the edge e_1 and replacing it with edge e_0, and call the resulting graph T_0'. In set notation,

$$T_0' = (T_0 - \{e_1\}) \cup \{e_0\}.$$

(b) Why is T_0' connected?

(c) Why is T_0' cycle-free?

(d) Why is T_0' a tree?

(e) What contradiction has been reached?

Exercise 4.13. Prove by induction on m that T_0 must contain the edges of $\mathfrak{G}_m(V)$.

Exercise 4.14. Why must the edges of T_0 be exactly those edges that occur in $\mathfrak{G}_c(V)$? Why is there only one solution to the minimum spanning tree problem given initial vertices with mutually different positive distances?

Exercise 4.15. Given a set V of n vertices in the plane (with mutually different positive distances), the Borůvka algorithm begins with an $n \times n$ distance matrix M as described in Exercise 4.3 (and also appearing at the beginning of his paper "On a Certain Minimal Problem").

(a) After the first iteration of Borůvka's algorithm, forming $\mathfrak{G}_0(V)$, why does the matrix M_1 have size at most $\frac{n}{2} \times \frac{n}{2}$?

(b) Why does Borůvka's algorithm produce a connected graph, $\mathfrak{G}_c(V)$, in at most $\log_2(n)$ iterations?

(c) Compute $\log_2(40)$ and compare this to the number of iterations needed to produce a connected graph for Borůvka's example of 40 points.

(d) Extra for Experts: Find a formula for the running time to completely execute Borůvka's algorithm. Hint: Consider the number of entries in the initial matrix M, namely n^2, and the number of iterations from part (b).

Exercise 4.16. (a) Write a computer program in the language of your choice that implements Borůvka's algorithm.

(b) Develop a graphic interface for your computer program that displays all of the initial points, and draws a separate picture for each iteration of Borůvka's algorithm.

References

[1] Barnett J., Bezhanishvili G., Leung H., Lodder J., Pengelley D., Pivkina I., Ranjan D., *Learning Discrete Mathematics and Computer Science via Primary Historical Sources*, `http://www.cs.nmsu.edu/historical-projects/`.

[2] Borůvka, O., "O jistém problému minimálnim," (On a Certain Minimal Problem), *Práce Moravské Přídovědecké Spolecnosti v Brně*, **3** (1926), 37–58.

[3] Borůvka, O., "Přispěvek k řešení otázky ekonomické stavby elektrovodních sítí" (A Contribution to the Solution of a Problem on the Economical Construction of Power Networks), *Elecktronický obzor*, **15** (1926), 153–154.

[4] Borůvka, O., "Několik vzpomínek na matematický žívot v Brně," (Some Memories of my Mathematical Life in Brno), *Pokroky Mat., Fyz. a Astr.*, **22** (1977), 91–99.

[5] Borůvka, O., *Grundlagen der Gruppoid und Gruppentheorie*, Deutscher Verlag der Wissenschaften, Berlin, 1960.

[6] Borůvka, O., *Lineare Differentialtransformationen 2. Ordnung*, Deutscher Verlag der Wissenschaften, Berlin, 1967.

[7] Borůvka, O., *Linear Differential Transformations of the Second Order*, (Arscott, F.M., trans.) English Universities Press, London, 1971.

[8] Borůvka, O., *Foundations of the Theory of Groupoids and Groups*, (Borukova, M. trans.), Wiley, New York, 1976.

[9] Cayley, A., "On the Theory of the Analytical Forms Called Trees," *Philosophical Magazine*, 4, **13** (1857), 172–176.

[10] Cayley, A., "A Theorem on Trees," *Quarterly Journal of Pure and Applied Mathematics*, **23** (1889), 376–378.

[11] Dziobek, O., "Eine Formel der Substitutionstheorie," *Sitzungsberichte der Berliner Mathematischen Gesellschaft*, **17** (1917), 64–67.

[12] Graham, R.L., Hell, P., "On the History of the Minimum Spanning Tree Problem," *Annals of the History of Computing*, **7**, 1 (1985), 43–57.

[13] Katz, V., *A History of Mathematics: An Introduction*, second ed., Addison Wesley, New York, 1998.

[14] Kirchhoff, G.R., "Über die Auflösung Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanishcer Strömre geführt wird," *Annalen der Physik und Chemie*, **72** (1847), 497–508.

[15] Kirchhoff, G.R., "Über die Auflösung Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanishcer Strömre geführt wird," translation in *Graph Theory 1736–1936*, Biggs, N. L., Lloyd, E. K., Wilson, R. J. (editors), Clarendon Press, Oxford, 1976.

[16] Kruskal, J.B., "On the shortest spanning subtree of a graph and the travelling salesman problem," *Proceedings of the American Mathematical Society*, **7**, 1 (1956), 48–50.

[17] Lawler, Eugene L., *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

[18] Nešetřil, J., Milková, E., Nešetřilová, H., "Otakar Borůvka on Minimum Spanning Tree Problem: Translation of Both 1926 Papers," *Discrete Mathematics*, **233**, 1 (2001), 3–36.

[19] Prim, R.C., "The shortest connecting network and some generalizations," *Bell Systems Technology Journal*, **36** (1957), 1389–1401.

[20] Prüfer, H. "Neuer Beweis eines Satzes über Permutationen," *Archiv der Mathematik und Physik*, 3, **27** (1918), 142–144.

[21] Prüfer, H. "Neuer Beweis eines Satzes über Permutationen," translation in *Graph Theory 1736–1936*, Biggs, N. L., Lloyd, E. K., Wilson, R. J. (editors), Clarendon Press, Oxford, 1976.

[22] O'Connor J.J., Robertson, E.F., "Ernst Paul Heinz Prüfer," `www-history.mcs.st-and.ac.uk/Mathematicians/Prufer.html`.

[23] Sylvester, J.J., "Chemistry and Algebra," *Nature*, **17** (1877–78), 284.

[24] Veblen, O. *Analysis Situs*, American Mathematical Society, New York, 1922.

Notes to the Instructor

The project is designed to motivate the modern definition of a "tree" found in textbooks covering graph theory, and then offer several applications of trees as well as one of the first algorithms for finding a minimal spanning tree. The term "tree" arises from the work of Arthur Cayley, whose enumeration of trees is discussed in short excerpts from "On the Theory of the Analytical Forms Called Trees" [9] and "A Theorem on Trees" [10]. This is contrasted with Heinz Prüfer's counting of trees, although the word "tree" never appears in his work. Prüfer introduces the material via an applied problem, namely the counting of all possible railway networks satisfying certain properties. In hindsight, each of these networks represents a "labeled tree." Finally an efficient algorithm for finding a minimal spanning tree is studied from the original work of Otakar Borůvka, who likewise discusses the problem without use of the term "tree." Borůvka sought the most economical construction of an electrical power network across the rural region of Southern Moravia, now part of the Czech Republic. This problem can be understood today as finding the tree of shortest total edge length from all possible n^{n-2} labeled trees on n towns. The number n^{n-2} here agrees with "Cayley's formula" for the number of labeled trees on n vertices. A prerequisite for the project is an introductory course in discrete mathematics covering such topics as induction and the concept of a one-to-one correspondence, needed for the Prüfer source. Also, Cayley's first paper "On the Theory of the Analytical Forms Called Trees" [9] requires knowledge of partial derivatives, although this is not necessary for an understanding of "Cayley's formula" found in the second paper "A Theorem on Trees" [10]. To cover the project in its entirety, allow about four weeks.

This curricular module requires no prior knowledge of graph theory. It is designed primarily for an advanced undergraduate course in combinatorics, graph theory or possibly algorithm design. The instructor may wish to cover the three main sections, highlighting source material form Cayley, Prüfer and Borůvka. Exercises from the Cayley source offer insight into the discovery of "Cayley's formula," while Prüfer's work provides a clever and geometrically appealing proof of this formula. Finally, Borůvka's algorithm offers a useful application of trees to solve a compelling problem in an efficient manner. For instructors seeking a hurried coverage of the project, study of Cayley's first paper "On the Theory of the Analytical Forms Called Trees" [9] could be replaced with the simple statement that Cayley introduces the term "tree" in this paper. The work of Prüfer and Borůvka can be read independently of Cayley, although "Cayley's formula" would appear without its historical origin, if the reading from the second paper "A Theorem on Trees" [10] is skipped. There are many exercises, some requiring simple recognition of a definition, and some requiring proofs of statements in the readings. The instructor is asked to work though the details of an exercise before assignment. Also, some knowledge of computer science is needed for exercises asking students to code a particular algorithm or exercises asking students to compare iteration with recursion. These may be omitted or further developed at the instructor's discretion.