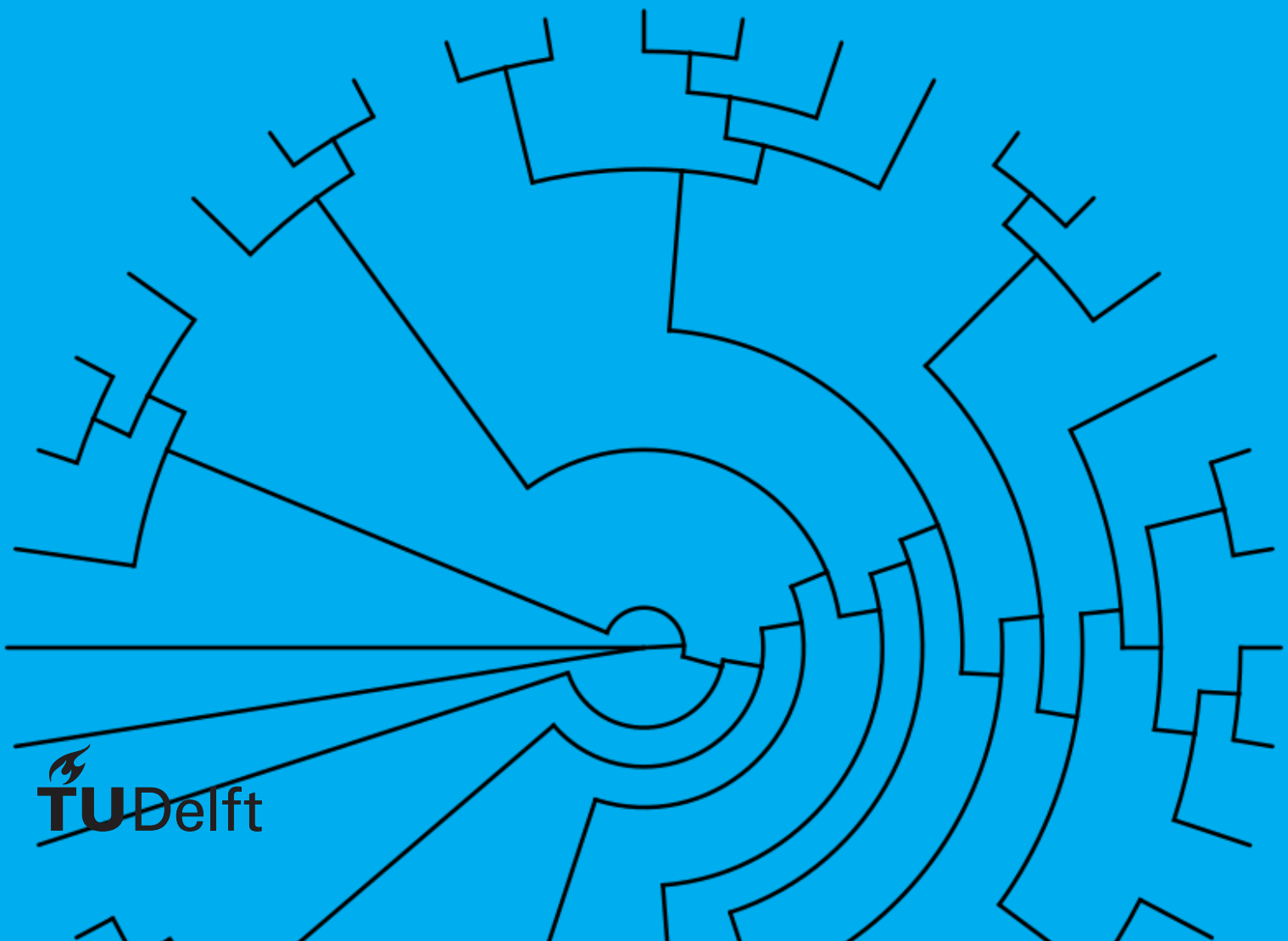
# Spectra of rooted binary phylogenetic trees

## Exploring the link between tree topology and eigenvalue patterns

N.V. Baars

Bachelor Thesis
Applied Mathematics
Delft University of Technology



TU Delft

# Spectra of rooted binary phylogenetic trees

## Exploring the link between tree topology and eigenvalue patterns

by

N.V. Baars

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended on Monday July 7, 2025 at 01:00 PM.


| | | |
|---|---|---|
| Student number: | 5637317 | |
| Project duration: | April 22, 2025 – July 7, 2025 | |
| Thesis committee: | Dr. Y. (Yuki) Murakami, | TU Delft, supervisor |
| | Dr. ir. W.G.M. (Wolter) Groenevelt, | TU Delft, assessment committee member |

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.


TU Delft

# Lay summary

Phylogenetic analysis has been used for decades to reconstruct the evolution of genes, individuals and species. Phylogenetic trees are used to display these evolutionary relationships. Comparison of trees can help identify evolutionary differences and prioritize conservations based on evolutionary value, which is interesting for many areas of biological research. However, since trees can become very large and complicated, it is hard to compare them. In spectral analysis, we store evolutionary distances between two current species in mathematical objects called matrices. Then, we can compare different trees by looking at their spectra - a set of numbers called eigenvalues, which are derived from these evolutionary distances. It already has been discovered that spectra cannot distinguish all trees, however, we show that some eigenvalues can reveal certain (sub)structures in phylogenetic trees. Take for example the eigenvalue $-2$, which tells us there are two current species sharing a parent. Besides, we will prove that other types of tree structures, such as perfectly symmetrical ones, have their own spectrum as well. Moreover, we show the condition under which eigenvalues of subtrees appear in the spectra of the entire tree. Although the spectra do not reveal the full structure of a phylogenetic tree, this approach could help researchers compare evolutionary trees much more efficiently.

# Summary

Phylogenetic trees have been used for decades to visualize evolutionary relationships graphically. Comparing topologies of trees is essential to many research areas of biology, but is complicated due to their combinatorial nature and the number of possible topologies that increases with the number of species. We will focus on binary rooted phylogenetic trees with unit edge length. Comparison can be facilitated by investigating the spectra - the set of eigenvalues - of the pairwise distance matrices of these trees. A pair of distinct trees on 17 leaves with equal spectrum already showed that spectra are not unique for the topology of the tree, however, they reveal some (sub-)structures.

In this thesis, we show that eigenvalue $-2$ with corresponding eigenvector $\vec{v} = (-e_i + e_j)$ reveals the presence of a cherry (two current species sharing a parent). Moreover, we prove that perfectly balanced trees have negative spectrum of the closed-form $-2(2^k - 1)$, where $k$ is a number between 1 and the height of the tree. In addition, we show that an eigenvalue $\lambda$ of a submatrix appears in the spectrum of the full matrix, as long as the part of the matrix linking the subtree to the rest of the tree is orthogonal to the submatrix's eigenvector corresponding to $\lambda$. The remaining eigenvalues of the full matrix can be computed using Schur's formula. Finally, we combine all these results and explain the spectral equivalence in the pair of distinct trees on 17 leaves. We observe that other eigenvalues that appear in this spectrum might reveal another type of subtree.

# Contents

# 1 Introduction

In recent decades, phylogenetic analysis has become a crucial tool for reconstructing the evolutionary history of genes, individuals and species [6]. Phylogenetic trees provide a standard graphical representation of these evolutionary relationships. The branch lengths between species in such trees represent the divergence time of the evolutionary change. The branching pattern of the tree is known as its topology. In Figure 1.1, a simplified phylogenetic tree with humans as one of the species is displayed [1]. Here, you can see that humans share a closer ancestor with chimpanzees than with gorillas.

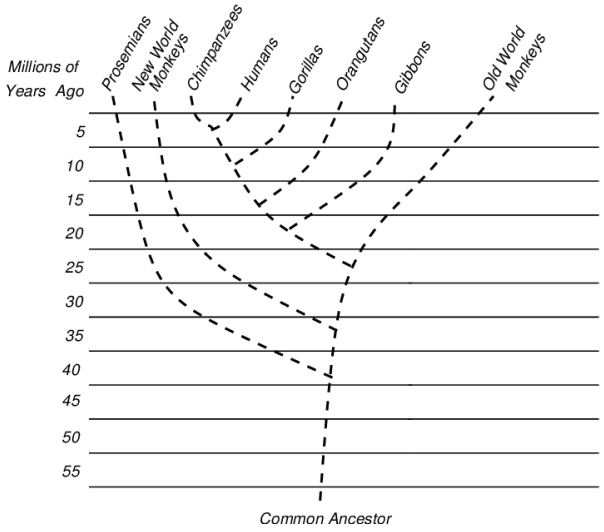



Figure 1.1: Simplified human evolution depicted in a phylogenetic tree [1].

The applications of phylogenetic trees are wide-ranging: from studying the origin and epidemiology of human diseases such as COVID-19 [9], to creating strategies to conserve biological diversity. Vézquez et al. state that, to conserve biodiversity, “it is necessary not only to maximize the number of taxa that are saved today, but also to guarantee the maintenance of high levels of biological diversity in the future. A recent analysis argues that, to achieve this, consideration of phylogeny is essential." [15]

For these applications, comparing the topologies of trees and summarizing their properties is of vital importance. However, analyzing and comparing phylogenetic trees is not straightforward. Comparison is complicated by the combinatorial nature of the trees and the number of possible topologies that increases rapidly with the number of species [8]. To address these challenges, mathematicians have combined linear algebra and graph theory to capture key aspects of a tree in a more manageable form. Expressing phylogenetic trees in mathematical objects, such as matrices, transforms them from intuitive biological diagrams to precise, analyzable mathematical structures.

There exist several ways to encode a tree as a matrix. Throughout this thesis, we focus on binary rooted phylogenetic trees with unit edge lengths, and we only consider their pairwise distance matrices. Pairwise distance matrices describe the evolutionary distance between two *current* species (taxa), which are the leaves in the tree. Therefore, only labeling of the leaves is required. Since we are interested in the structure of phylogenetic trees, rather than the identities of the taxa, we want to look at features of the matrix that will not change after relabeling. Such a feature is the spectrum of the matrix, in other words, the set of eigenvalues. If tree topologies can be revealed through the spectrum of its pairwise distance matrix, comparison and pattern detecting will be significantly more efficient. This is because instead of working with large matrices that grow with the number of species, we only need to work with a much smaller set of numbers.

A previous investigation by Graham and Lovász used distance matrices, which describe the distances between all vertices instead of only the leaves. They showed that we can read off structural details of a tree, including subtree counts, directly from the coefficients of the characteristic polynomial of its distance matrix [4]. Since this polynomial determines the eigenvalues, their result highlights how tree topology is reflected algebraically. However, less is known about how tree patterns manifest in the spectra of pairwise distance matrices, which are principal submatrices of distance matrices. Below, we state what has been discovered thus far and how we will expand these results.

Unfortunately, Matsen and Evans already discovered that the spectra of pairwise distance matrices of binary rooted phylogenetic trees do not uniquely determine the structure of the phylogenetic tree; Figure 1.2 shows two distinct trees on seventeen taxa that share the same spectrum. More specifically, they discovered that the number of trees that is uniquely determined by its spectrum goes to zero as the number of leaves in the tree goes to infinity [10].

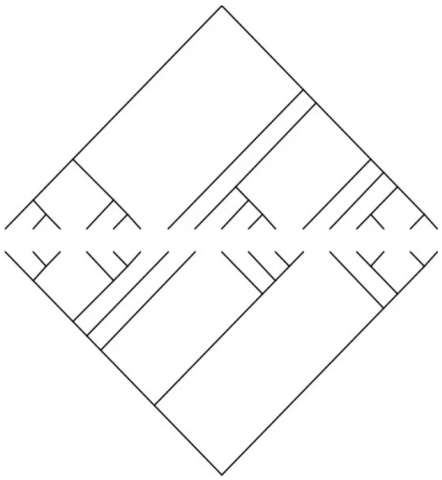


Figure 1.2: Two different phylogenetic trees with the same pairwise distance spectrum.

Despite this non-uniqueness, structural features of trees are still visible in their spectra. Singh et al. showed that the spectra of balanced trees (binary ultrametric trees) only contain negative eigenvalues of a specific form [14]. However, in their paper, no proof was given. In this thesis, we will give a proof on this closed-form spectrum, using linear algebra.

Moreover, De Ponte and De Campos investigated under what conditions eigenvalues of submatrices - such as those corresponding to subtrees - can be inherited by the full matrix [12]. Therefore, by examining spectra of subtrees, we deduce partial spectral information about the entire tree. In this paper, we will discuss the kind of subtree that always meets these conditions.

Lastly, we will combine these results and give an explanation on why the phylogenetic trees in Figure 1.2 share the same spectrum, which is an extension of the result of Matsen and Evans.

The outline of this report is as follows. In the next chapter, we will start with some background information on phylogenetic trees and spectra. In Chapter 3, we investigate the spectra of the trees in Figure 1.2 and discuss the questions they raise. In Sections 3.1, 3.2 and 3.3 we address one of these questions, such as the inheritance of eigenvalues from submatrices, eigenvalues of cherries and spectra of perfectly balanced trees.

In Section 3.4, we will combine the results and explain the spectral equivalence of the trees in Figure 1.2. In Section 3.5, we find other eigenvalues that might reveal a specific type of subtree. Finally, in Chapter 4, we give an overall conclusion and discuss some conjectures and directions for future work.

# 2

# Preliminaries

In this chapter, we state some preliminaries and definitions which are needed for the rest of this thesis. Mainly, results from graph theory and linear algebra are included, which can be found in [2, 3, 7, 13, 16].

## 2.1. Graphs

Before introducing the trees we consider in this thesis, it is useful to define graphs in general. A *graph* is a representation of a set of points and the way they are connected. Formally, a graph $G$ is a pair of sets $(V, E)$ where $V = V(G)$ is known as the set of *vertices*, the objects, and the set $E = E(G)$ is known as the set of *edges*, the connections between the objects. In Figure 2.1, the circles are the vertices, and the lines between these vertices are the edges. A *directed graph* is a graph in which the edges have a specific direction. The *underlying graph* of a directed graph is the graph obtained from 'removing the arrows'. Graphs that are not directed are either *undirected* or *semi-directed*. We will only consider directed graphs, hereafter 'graphs'.

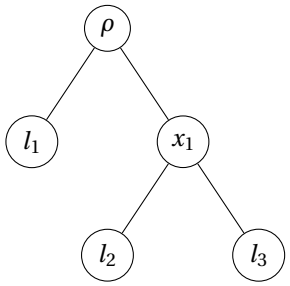



Figure 2.1: A graph with five vertices $\rho, x_1, l_1, l_2$ and $l_3$ and four edges.

If a graph $G$ contains an edge $e = ab$ from $a$ to $b$, then $e$ is *incident* with $a$ and $b$. Moreover, $a$ is said to be *adjacent to* $b$, and $b$ is said to be *adjacent from* $a$. The vertex $a$ is called the *initial vertex* of $e$, and $b$ is called the *terminal vertex* of $e$. The *in-degree* of $a$ is the number of edges with $a$ as terminal vertex, while the *out-degree* of $a$ is the number of edges with $a$ as initial vertex. The *degree* of a vertex $a$, denoted by $\deg(a)$, is the sum of its in-degree and out-degree. For example, in Figure 2.1, $\deg(\rho) = 2$.

A *path* between two vertices $a$ and $b$ is a sequence of edges that connects $a$ and $b$, such that no edge and vertex appear more than once. In this thesis, we only consider edges of length 1. Then, the *length* of a path from $a$ to $b$ is the number of edges in this path. A graph is *connected* if any set of two vertices in its underlying graph is connected by a path. If any set of two vertices in the graph itself is connected by a path, the graph is *strongly connected*. A graph contains a *cycle* if there is a path from a vertex $a$ to itself, as displayed in Figure 2.2.

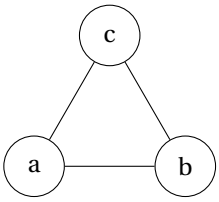



Figure 2.2: A cycle graph on three vertices $a, b$ and $c$.

## 2.2. (Phylogenetic) trees

In this thesis, we will look at a specific type of directed graphs, called directed trees. A *directed tree*, often referred to as $T$, is a weakly connected graph that does not contain any cycles. Directed trees have the property that any two vertices in the underlying graph are connected by a unique path. A *leaf* in a tree $T$ is a vertex of degree 1. Let the leaves of a directed tree on $n$ leaves be arbitrarily ordered by integers from 1 to $n$. Then, we denote the $i$th leaf by $l_i$. A vertex of $T$ that is not a leaf is called an *interior vertex* or *in-between vertex*. Similarly, if we order the interior vertices arbitrarily, we denote the $i$th interior vertex by $x_i$.

A directed *rooted tree* is a directed tree that has exactly one distinguished vertex called the *root*, which we denote by the letter $\rho$. For modeling evolutionary histories, as we do in this thesis, a rooted phylogenetic tree is used. A *rooted phylogenetic tree* $T$ is a directed rooted tree with no degree-two vertices, except (possibly) the root $\rho$ which has degree at least two. In such a tree, all edges are directed away from the root. Each edge represents an evolutionary transition, and the leaves represent the current species (*taxa*). We will only consider *binary* rooted phylogenetic trees, which means that there is either only a leaf or the root has degree 2 (in-degree 0 and out-degree 2) and every interior vertex has degree 3 (in-degree 1 and out-degree 2). Note that the graph in Figure 2.1 is a binary rooted phylogenetic tree with root $\rho$, interior vertex $x_1$ and leaf-set $\{l_1, l_2, l_3\}$.

Two distinct leaves of a binary rooted phylogenetic tree are said to form a *cherry* if they are adjacent from a common vertex. In Figure 2.1, $l_2$ and $l_3$ form a cherry. We say that $x_1$ is a *parent* of $l_2$ and $l_3$ and, conversely, $l_2$ and $l_3$ are *children* of $x_1$. Thus, children and parents are separated by 1 edge. The root $\rho$ is an *ancestor* to all leaves and in-between vertices. Similarly, all interior vertices and leaves are *descendants* of $\rho$.

We will take a deeper look at two types of binary rooted phylogenetic trees, the *caterpillars* and the *perfectly balanced trees*. The *depth* of a vertex is the number of edges in the path from the root to the vertex, and the *height* of a tree is the number of edges from the root to the deepest leaf.
A *rooted caterpillar* is a binary rooted phylogenetic tree where all vertices are either part of a central path or are adjacent from a vertex in the central path. In Figure 2.3 a caterpillar on 4 leaves is displayed.

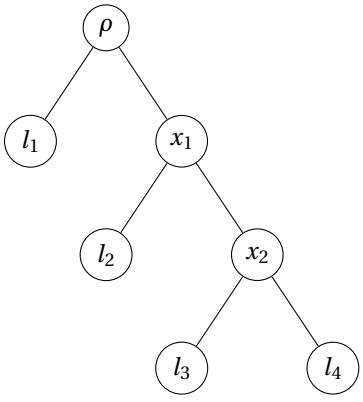



Figure 2.3: A rooted caterpillar on 4 leaves.

A *perfectly balanced tree* of height $h \geq 0$ is a binary rooted phylogenetic tree with $n = 2^h$ leaves, each of which is separated from the root by exactly $h$ edges. We will denote a perfectly balanced tree of height $h$ by $T_h$. A perfectly balanced tree of height 2 is displayed in Figure 2.4.

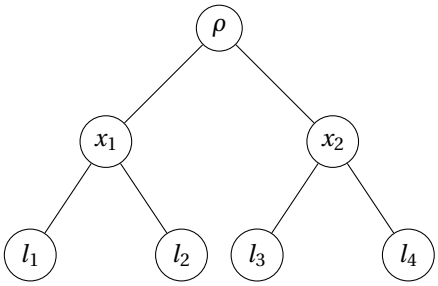



Figure 2.4: The perfectly balanced tree $T_2$ on 4 leaves.

In this thesis, we will frequently look at specific types of *subtrees*. A *subtree* $T' = (V', E')$ of a tree $T = (V, E)$ is a tree, such that $V'$ is a subset of $V$ and $E'$ is a subset of $E$. That is, each vertex in $V'$ is a vertex of $V$ and similarly, each edge of $E'$ is an edge of $E$. The subtree is called *induced* if, for all $a, b \in V'$ for which the edge $e = ab$ is in $E$, we have $e \in E'$ as well. Now we can introduce a commonly used term in this thesis, the *pendant subtree*. Let $T = (V, E)$ be a binary rooted phylogenetic tree. Pick any vertex $a$ and let $T' = (V', E')$ be the subtree induced by $a$ and all its descendants. Then $T'$ is a *pendant subtree*.

## 2.3. Matrices and their spectra

In this section, we will define spectra and give some properties of matrices that will be used in this thesis.

Let $M$ be an $n \times n$ matrix. A scalar $\lambda$ is an *eigenvalue* of a matrix $M$ if there is a non-zero vector $\vec{v} = [v_1 \cdots v_n]^T \in \mathbb{R}^n$ such that $M\vec{v} = \lambda\vec{v}$. The vector $\vec{v}$ is an *eigenvector* of $M$ corresponding to $\lambda$. The *standard unit vector*, denoted by $e_j$, is a vector with $v_j = 1$ and $v_i = 0$ for all $i \neq j$. The *span* of a set of vectors $V$, denoted by $\text{Span}(V)$, is the set of all their linear combinations.

Eigenvalues are solutions of the equation $p(\lambda) = \det(M - \lambda I) = 0$, also known as the *characteristic polynomial* of $D(T)$, when solving for $\lambda$. It can be factored as $p(\lambda) = (\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n)$, where each $\lambda_i$ is an eigenvalue of $M$. Note that $\det(M) = p(0) = (-1)^n \prod_{i=1}^{n} \lambda_i$. The *algebraic multiplicity* of an eigenvalue $\lambda_i$ is its multiplicity as a solution to the characteristic polynomial. The *spectrum* of a matrix $M$ is defined as the collection of its $k$ distinct eigenvalues $\lambda_i$, $i \in \{1, \ldots, k\}$, with their corresponding algebraic multiplicities $m_i$, denoted by $\text{Spec}(M) = \{(\lambda_1)^{m_1}, ..., (\lambda_k)^{m_k}\}$.

For small matrices, we can easily compute the eigenvalues by hand. The eigenvalues of a $2 \times 2$ matrix $M = \left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ can be computed by

$$\det(M - \lambda I) = \begin{vmatrix} a - \lambda & b \\ c & d - \lambda \end{vmatrix} = (a - \lambda)(d - \lambda) - bc = 0.$$

In Section 2.4, we will give a more detailed explanation on determinants of the matrices we consider in this thesis.

Specific types of matrices can have useful properties. Symmetric matrices, for example, always have real eigenvalues. That is, $\lambda_i \in \mathbb{R}$ for all $i$. In addition, eigenvectors of symmetric matrices are *pairwise orthogonal*, which means that for any pair of eigenvectors $\vec{v}$ and $\vec{u}$, we must have $\vec{v}^T\vec{u} = 0$. The eigenvectors are *pairwise orthonormal* if they are orthogonal and have norm 1. The *norm* of a vector $\vec{v}$ is defined as $||\vec{v}|| = \sqrt{v_1^2 + \cdots + v_n^2}$. Note that pairwise orthogonal vectors can be made orthonormal by multiplying each vector $\vec{v}$ by $\frac{1}{||\vec{v}||}$. An *orthogonal* matrix is a square matrix whose columns and rows are orthonormal vectors. We know that, for an orthogonal matrix $M$, $\det(M) = \pm 1$ and $M^{-1} = M^T$. Here, $M^{-1}$ is the *inverse* of $M$, which means that $MM^{-1} = M^{-1}M = I$.

Moreover, a matrix $M$ is *diagonalizable* if it can be rewritten as $U\Lambda_M U^T$. Here, $U = [\vec{v}_1 \cdots \vec{v}_n]$ is an orthogonal matrix with each $\vec{v}_i$ an eigenvector corresponding to eigenvalue $\lambda_i$, and $\Lambda_M = \text{diag}(\lambda_1, \ldots, \lambda_n)$. Symmetric matrices are always diagonalizable.

Performing a *similarity transformation* on a matrix $M$ means we reorder rows and the corresponding columns. For this, a *permutation matrix* $P$ is needed, which is an orthogonal and invertible matrix such that $B = PAP^T$, where $B$ is the new matrix with reordered rows and columns. Since

$$\det(B-\lambda I) = \det(PAP^T - \lambda I) = \det(PAP^T - \lambda PP^T) = \det(P(A-\lambda I)P^T) = \det(A-\lambda I),$$

the eigenvalues of $A$ and $B$ are equal.

Before we move on to the type of matrices we consider in this thesis, we state a few more (in)equalities that we use later on.

**Lemma 1.** *The trace of an $n \times n$ matrix $M$, the sum of its diagonal entries, equals the sum of its eigenvalues:* $\mathrm{tr}(M) = \sum_{i=1}^{n} M_{ii} = \sum_{i=1}^{n} \lambda_i$ *[7].*

**Lemma 2.** *Schur's formula states that the determinant of a matrix* $M = \left[\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right]$ *can be computed by* $\det(M) = \det(A)\det(D - CA^{-1}B)$ *if the submatrix $A$ is invertible [7].*

**Theorem 1.** *Cauchy's interlacing theorem states that, if* $M = \left[\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right]$ *is an $n \times n$ real symmetric matrix with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$, and $A$ is an $m \times m$ submatrix with eigenvalues by $\mu_1 \leq \cdots \mu_m$, then for all $k = 1, \ldots, m$, it holds that $\lambda_k \leq \mu_k \leq \lambda_{k+n-m}$ [7].*

## 2.4. Spectra of phylogenetic trees

Now that we introduced some basic definitions of binary rooted phylogenetic trees (henceforth referred to simply as 'trees') and gave some properties of matrices, we turn to their pairwise distance matrices and associated spectra.

Every tree $T$ on $n$ leaves can be represented by a matrix as follows. Let the leaves be arbitrarily ordered by integers from 1 to $n$. Then the *pairwise distance matrix* or *inter-taxa distance matrix* of $T$, denoted by $D(T)$, is an $n \times n$ matrix where the entry $D_{ij}$ represents the length of the *up-down path* between leaf $l_i$ and leaf $l_j$, known as the distance $d(l_i, l_j)$ [10]. The up-down path is the path from $l_i$ to $l_j$ via their *most recent common ancestor*, without taking into account the edges' directions. The *most recent common ancestor* of two leaves $l_i$ and $l_j$, or the *lowest common ancestor*, is an ancestor to both leaves such that it is lowest. For example, in Figure 2.3, $x_1$ is the lowest common ancestor to $l_2$ and $l_4$. Note that $D(T)$ is a real symmetric matrix with zeros on the diagonal. In other words, $d(l_i, l_j) = d(l_j, l_i)$ and $d(l_i, l_i) = 0$. Hence, for a tree $T$ on $n$ leaves,

$$D(T) = \begin{bmatrix} 0 & d(l_1,l_2) & \ldots & d(l_1,l_n) \\ d(l_2,l_1) & 0 & \ldots & d(l_2,l_n) \\ \vdots & \vdots & \ddots & \vdots \\ d(l_n,l_1) & d(l_n,l_2) & \ldots & 0 \end{bmatrix}.$$

Because of the symmetry, we can use the properties for symmetric matrices stated in Section 2.3.

We are interested in the eigenvalues of these matrices. For a $2 \times 2$ pairwise distance matrix $D(T) = \left[\begin{smallmatrix} 0 & d(l_1,l_2) \\ d(l_2,l_1) & 0 \end{smallmatrix}\right]$, the eigenvalues can be computed by

$$\det(D(T) - \lambda I) = \begin{vmatrix} -\lambda & d(l_1,l_2) \\ d(l_2,l_1) & -\lambda \end{vmatrix} = \lambda^2 - d(l_1,l_2)d(l_2,l_1) = \lambda^2 - d(l_1,l_2)^2,$$

which results in eigenvalues $\lambda_{1,2} = \pm d(l_1, l_2)$. For larger matrices, computing determinants is very time-consuming. Therefore, in Appendix A, a Python code for computing eigenvalues can be found. If interested, the characteristic polynomial for $3 \times 3$ and $n \times n$ pairwise distance matrices in general can be found in Appendix C.

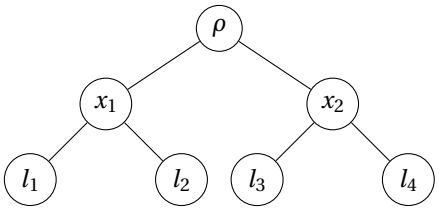
Note that relabeling the leaves in the tree $T$ is equivalent to performing a similarity transformation on the matrix $D(T)$. Since performing such a transformation will not change the spectrum, we can order the leaves arbitrarily. That is, the spectrum of $D(T)$ is *invariant* under relabeling.

Since a pairwise distance matrix has zeros on the diagonal, we can apply Lemma 1 to obtain the following corollary:

**Corollary 1.** *For a pairwise distance matrix $D(T)$,* $\mathrm{tr}(D(T)) = \sum_{i=1}^{n} D_{ii}(T) = \sum_{i=1}^{n} \lambda_i = 0$.

Now we will take a look at a small example of a tree $T$ on 4 leaves.

**Example 1.** Here, we consider the perfectly balanced tree $T_2$ displayed in Figure 2.4. We will display this figure below again for convenience.




The pairwise distance matrix corresponding to this tree is given by:

$$D(T_2) = \begin{bmatrix} 0 & d(l_1,l_2) & d(l_1,l_3) & d(l_1,l_4) \\ d(l_2,l_1) & 0 & d(l_2,l_3) & d(l_2,l_4) \\ d(l_3,l_1) & d(l_3,l_2) & 0 & d(l_3,l_4) \\ d(l_4,l_1) & d(l_4,l_2) & d(l_4,l_3) & 0 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 4 & 4 \\ 2 & 0 & 4 & 4 \\ 4 & 4 & 0 & 2 \\ 4 & 4 & 2 & 0 \end{bmatrix}.$$

We can compute the eigenvalues of $D(T_2)$ as follows:

$$\det(D(T_2) - \lambda I) = \begin{vmatrix} -\lambda & 2 & 4 & 4 \\ 2 & -\lambda & 4 & 4 \\ 4 & 4 & -\lambda & 2 \\ 4 & 4 & 2 & -\lambda \end{vmatrix} = \lambda^4 - 72\lambda^2 - 256\lambda - 240 = 0,$$

which has solutions $\lambda_1 = -2, \lambda_2 = -2, \lambda_3 = -6$ and $\lambda_4 = 10$. Corresponding eigenvectors are $\vec{v}_1 = \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, $\vec{v}_2 = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix}$, $\vec{v}_3 = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$ and $\vec{v}_4 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$. The spectrum of $D(T)$ is given by $\{(-2)^2, (-6)^1, (10)^1\}$.

# 3

# Uncovering tree structures through their spectra

In this chapter, we will explore how specific tree patterns manifest in the spectrum of pairwise distance matrices of binary rooted phylogenetic trees, referred to as 'trees'. As we already mentioned in Chapter 1, Matsen and Evans found two different trees on 17 leaves, which we will denote by $T_{17,1}$ and $T_{17,2}$, with the same set of eigenvalues [10]. These trees are displayed in Figure 3.1 and 3.2.

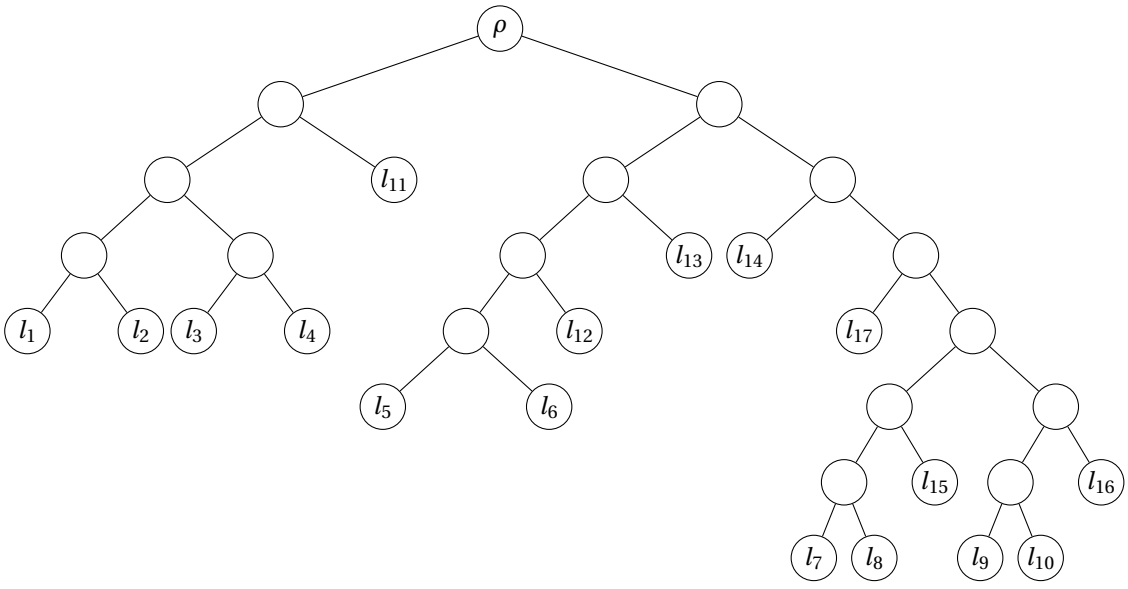



Figure 3.1: The phylogenetic tree $T_{17,1}$ that is not uniquely determined by its spectrum. The spectrum is: $\{(-40.450)^1, (-22.912)^1, (-11.123)^1, (-6.949)^1, (-6)^1, (-3.938)^1, (-3.579)^1, (-3.174)^1, (-2.877)^1, (-2.742)^1, (-2.658)^1, (-2)^5, (116.403)^1\}$, where each non-integer eigenvalue is rounded to three decimals.

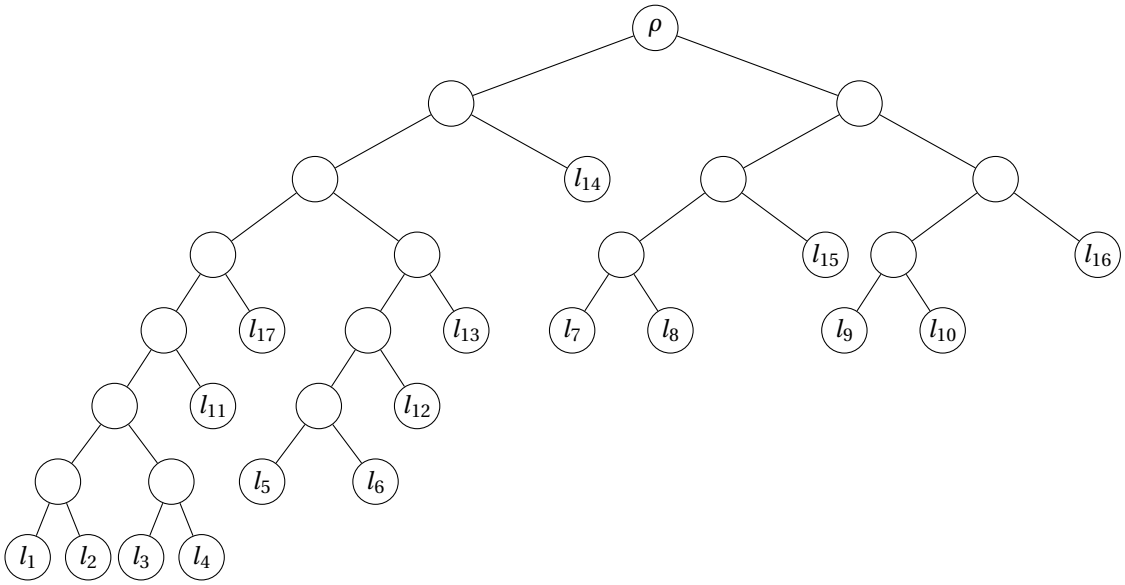



Figure 3.2: The phylogenetic tree $T_{17,2}$ that is not uniquely determined by its spectrum. The spectrum is: $\{(-40.450)^1, (-22.912)^1, (-11.123)^1, (-6.949)^1, (-6)^1, (-3.938)^1, (-3.579)^1, (-3.174)^1, (-2.877)^1, (-2.742)^1, (-2.658)^1, (-2)^5, (116.403)^1\}$, where each non-integer eigenvalue is rounded to three decimals.

The pairwise distance matrices corresponding to these trees with the given labeling are given below.

$$D(T_{17,1}) = \begin{bmatrix}
0 & 2 & 4 & 4 & 9 & 9 & 11 & 11 & 11 & 11 & 4 & 8 & 7 & 7 & 10 & 10 & 8 \\
2 & 0 & 4 & 4 & 9 & 9 & 11 & 11 & 11 & 11 & 4 & 8 & 7 & 7 & 10 & 10 & 8 \\
4 & 4 & 0 & 2 & 9 & 9 & 11 & 11 & 11 & 11 & 4 & 8 & 7 & 7 & 10 & 10 & 8 \\
4 & 4 & 2 & 0 & 9 & 9 & 11 & 11 & 11 & 11 & 4 & 8 & 7 & 7 & 10 & 10 & 8 \\
9 & 9 & 9 & 9 & 0 & 2 & 10 & 10 & 10 & 10 & 7 & 3 & 4 & 6 & 9 & 9 & 7 \\
9 & 9 & 9 & 9 & 2 & 0 & 10 & 10 & 10 & 10 & 7 & 3 & 4 & 6 & 9 & 9 & 7 \\
11 & 11 & 11 & 11 & 10 & 10 & 0 & 2 & 6 & 6 & 9 & 9 & 8 & 6 & 3 & 5 & 5 \\
11 & 11 & 11 & 11 & 10 & 10 & 2 & 0 & 6 & 6 & 9 & 9 & 8 & 6 & 3 & 5 & 5 \\
11 & 11 & 11 & 11 & 10 & 10 & 6 & 6 & 0 & 2 & 9 & 9 & 8 & 6 & 5 & 3 & 5 \\
11 & 11 & 11 & 11 & 10 & 10 & 6 & 6 & 2 & 0 & 9 & 9 & 8 & 6 & 5 & 3 & 5 \\
4 & 4 & 4 & 4 & 7 & 7 & 9 & 9 & 9 & 9 & 0 & 6 & 5 & 5 & 8 & 8 & 6 \\
8 & 8 & 8 & 8 & 3 & 3 & 9 & 9 & 9 & 9 & 6 & 0 & 3 & 5 & 8 & 8 & 6 \\
7 & 7 & 7 & 7 & 4 & 4 & 8 & 8 & 8 & 8 & 5 & 3 & 0 & 4 & 7 & 7 & 5 \\
7 & 7 & 7 & 7 & 6 & 6 & 6 & 6 & 6 & 6 & 5 & 5 & 4 & 0 & 5 & 5 & 3 \\
10 & 10 & 10 & 10 & 9 & 9 & 3 & 3 & 5 & 5 & 8 & 8 & 7 & 5 & 0 & 4 & 4 \\
10 & 10 & 10 & 10 & 9 & 9 & 5 & 5 & 3 & 3 & 8 & 8 & 7 & 5 & 4 & 0 & 4 \\
8 & 8 & 8 & 8 & 7 & 7 & 5 & 5 & 5 & 5 & 6 & 6 & 5 & 3 & 4 & 4 & 0
\end{bmatrix}$$

$$D(T_{17,2}) = \begin{bmatrix}
0 & 2 & 4 & 4 & 9 & 9 & 11 & 11 & 11 & 11 & 4 & 8 & 7 & 7 & 10 & 10 & 5 \\
2 & 0 & 4 & 4 & 9 & 9 & 11 & 11 & 11 & 11 & 4 & 8 & 7 & 7 & 10 & 10 & 5 \\
4 & 4 & 0 & 2 & 9 & 9 & 11 & 11 & 11 & 11 & 4 & 8 & 7 & 7 & 10 & 10 & 5 \\
4 & 4 & 2 & 0 & 9 & 9 & 11 & 11 & 11 & 11 & 4 & 8 & 7 & 7 & 10 & 10 & 5 \\
9 & 9 & 9 & 9 & 0 & 2 & 10 & 10 & 10 & 10 & 7 & 3 & 4 & 6 & 9 & 9 & 6 \\
9 & 9 & 9 & 9 & 2 & 0 & 10 & 10 & 10 & 10 & 7 & 3 & 4 & 6 & 9 & 9 & 6 \\
11 & 11 & 11 & 11 & 10 & 10 & 0 & 2 & 6 & 6 & 9 & 9 & 8 & 6 & 3 & 5 & 8 \\
11 & 11 & 11 & 11 & 10 & 10 & 2 & 0 & 6 & 6 & 9 & 9 & 8 & 6 & 3 & 5 & 8 \\
11 & 11 & 11 & 11 & 10 & 10 & 6 & 6 & 0 & 2 & 9 & 9 & 8 & 6 & 5 & 3 & 8 \\
11 & 11 & 11 & 11 & 10 & 10 & 6 & 6 & 2 & 0 & 9 & 9 & 8 & 6 & 5 & 3 & 8 \\
4 & 4 & 4 & 4 & 7 & 7 & 9 & 9 & 9 & 9 & 0 & 6 & 5 & 5 & 8 & 8 & 3 \\
8 & 8 & 8 & 8 & 3 & 3 & 9 & 9 & 9 & 9 & 6 & 0 & 3 & 5 & 8 & 8 & 5 \\
7 & 7 & 7 & 7 & 4 & 4 & 8 & 8 & 8 & 8 & 5 & 3 & 0 & 4 & 7 & 7 & 4 \\
7 & 7 & 7 & 7 & 6 & 6 & 6 & 6 & 6 & 6 & 5 & 5 & 4 & 0 & 5 & 5 & 4 \\
10 & 10 & 10 & 10 & 9 & 9 & 3 & 3 & 5 & 5 & 8 & 8 & 7 & 5 & 0 & 4 & 7 \\
10 & 10 & 10 & 10 & 9 & 9 & 5 & 5 & 3 & 3 & 8 & 8 & 7 & 5 & 4 & 0 & 7 \\
5 & 5 & 5 & 5 & 6 & 6 & 8 & 8 & 8 & 8 & 3 & 5 & 4 & 4 & 7 & 7 & 0
\end{bmatrix}$$

Now that we know that trees are not uniquely determined by their spectra, we want to discover to what extent the spectrum does reveal the topology of the tree. In these trees on 17 leaves, we observe a few remarkable

things:

- With this leaf-labeling, we see that the top-left $16 \times 16$ submatrix of both matrices is exactly the same. Therefore, in Section 3.1, we will discover what spectra of submatrices (subtrees) can tell us about the spectra of the full matrix.
- We see that $-2$ appears five times in the spectrum of $D(T_{17,1})$ and $D(T_{17,2})$. Besides, we see that both trees have exactly five cherries. Therefore, in Section 3.2, we want to discover if the eigenvalue $-2$ always corresponds to a cherry. Note that a cherry is a perfectly balanced tree of height 1. Therefore, in Section 3.3, we will take a further look at spectra of perfectly balanced (sub)trees of general height.
- In Lemma 4 of the paper of Matsen and Evans, a condition is mentioned for two trees to have the same spectrum. However, the trees $T_{17,1}$ and $T_{17,2}$ do not seem to meet this condition. Therefore, in Section 3.4, we will combine our results and give a sufficient explanation to the spectral equivalence of $T_{17,1}$ and $T_{17,2}$.

Lastly, in Section 3.5, we will look at negative integer eigenvalues of trees in general and find an example of eigenvalues that might reveal another type of subtree.

## 3.1. Extracting spectral information via submatrices

De Ponte and De Campos already discovered that the order of characteristic polynomials can be reduced, given the eigenvalues and eigenvectors of a submatrix [12]. In Lemma 3, we will state when eigenvalues of a submatrix of a pairwise distance matrix specifically can be inherited by the full matrix. In Lemma 4, we will give the reduced characteristic polynomial, which is a direct result of De Ponte and De Campos. For both lemmas, we will give our own, but similar proof.

**Lemma 3.** *Let $T$ be a tree on $n$ leaves with pendant subtree $T'$ on $m$ leaves. Label the leaves such that $D(T) = \begin{bmatrix} A & B & C \\ B^T & D(T') & E \\ C^T & E^T & F \end{bmatrix}$, where $D(T') \in \mathbb{R}^{m \times m}$ is the pairwise distance matrix of $T'$.*

*Then $\lambda$ is an eigenvalue of $D(T)$ with eigenvector $\begin{bmatrix} \vec{0} \\ \vec{u} \\ \vec{0} \end{bmatrix}$ if and only if $\lambda$ is an eigenvalue of $D(T')$ with eigenvector $\vec{u} \in \mathbb{R}^m$ and $B\vec{u} = \vec{0}$ and $E^T\vec{u} = \vec{0}$.*

*Proof.* For the first direction, suppose $D(T)$ has an eigenvalue $\lambda$ with eigenvector $\begin{bmatrix} \vec{0} \\ \vec{u} \\ \vec{0} \end{bmatrix}$. Then we know that $D(T)\begin{bmatrix} \vec{0} \\ \vec{u} \\ \vec{0} \end{bmatrix} = \lambda \begin{bmatrix} \vec{0} \\ \vec{u} \\ \vec{0} \end{bmatrix}$. Rewriting this gives us

$$\begin{bmatrix} A & B & C \\ B^T & D(T') & E \\ C^T & E^T & F \end{bmatrix} \begin{bmatrix} \vec{0} \\ \vec{u} \\ \vec{0} \end{bmatrix} = \lambda \begin{bmatrix} \vec{0} \\ \vec{u} \\ \vec{0} \end{bmatrix}$$

which leads to

$$A\vec{0} + B\vec{u} + C\vec{0} = \lambda\vec{0} \tag{3.1}$$

$$B^T\vec{0} + D(T')\vec{u} + E\vec{0} = \lambda\vec{u} \tag{3.2}$$

$$C^T\vec{0} + E^T\vec{u} + F\vec{0} = \lambda\vec{0} \tag{3.3}$$

Equation 3.1 shows us that $B\vec{u} = \vec{0}$. Equation 3.2 shows that $D(T')\vec{u} = \lambda\vec{u}$, which means $\lambda$ is an eigenvalue of $D(T')$ with eigenvector $\vec{u}$. Equation 3.3 implies $E^T\vec{u} = \vec{0}$.

For the opposite direction, suppose $\lambda$ is an eigenvalue of $D(T')$ with eigenvector $\vec{u}$ and $B\vec{u} = \vec{0}$ and $E^T\vec{u} = \vec{0}$. Then we know that $D(T')\vec{u} = \lambda\vec{u}$. We wish to show that $\lambda$ is an eigenvalue of $D(T)$ with corresponding eigenvector $\begin{bmatrix} \vec{0} \\ \vec{u} \\ \vec{0} \end{bmatrix}$. This follows from:

$$D(T)\begin{bmatrix} \vec{0} \\ \vec{u} \\ \vec{0} \end{bmatrix} = \begin{bmatrix} A & B & C \\ B^T & D(T') & E \\ C^T & E^T & F \end{bmatrix}\begin{bmatrix} \vec{0} \\ \vec{u} \\ \vec{0} \end{bmatrix} = \begin{bmatrix} A\vec{0} + B\vec{u} + C\vec{0} \\ B^T\vec{0} + D(T')\vec{u} + E\vec{0} \\ C^T\vec{0} + E^T\vec{u} + F\vec{0} \end{bmatrix} = \begin{bmatrix} \vec{0} \\ D(T')\vec{u} \\ \vec{0} \end{bmatrix} = \begin{bmatrix} \vec{0} \\ \lambda\vec{u} \\ \vec{0} \end{bmatrix} = \lambda\begin{bmatrix} \vec{0} \\ \vec{u} \\ \vec{0} \end{bmatrix}.$$

Hence, $\lambda$ is an eigenvalue of $D(T)$ with eigenvector $\begin{bmatrix} \vec{0} \\ \vec{u} \\ \vec{0} \end{bmatrix}$. □

The remaining eigenvalues, the ones that are not in the spectrum of a submatrix, can be computed as follows.

**Lemma 4.** *Let $T$ be a tree on $n+m$ leaves. Let $D(T) = \left[\begin{smallmatrix} A & B \\ B^T & C \end{smallmatrix}\right]$ be the pairwise distance matrix of $T$, where $A \in \mathbb{R}^{n\times n}$ is diagonalizable as $A = U\Lambda_A U^T$. Then the eigenvalues of $D(T)$ that are not in the spectrum of $A$ can be computed using*

$$\det(D(T) - \lambda I_{n+m}) = \left(\prod_{j=1}^{n}(\lambda_j - \lambda)\right) \cdot \det\left(C - \lambda I_m - \sum_{j=1}^{n} \frac{(\vec{u}_j{}^T B)^2}{\lambda_j - \lambda}\right).$$

*Proof.* The following matrix is needed for computing the eigenvalues of $D(T)$:

$$D(T) - \lambda I_{n+m} = \begin{bmatrix} A - \lambda I_n & B \\ B^T & C - \lambda I_m \end{bmatrix}.$$

If $\lambda$ is not an eigenvalue of $A$, then $(A - \lambda I_n)$ is invertible and according to Lemma 2,

$$\det(D(T) - \lambda I) = \det(A - \lambda I_n)\det(C - \lambda I_m - B^T(A - \lambda I_n)^{-1}B). \tag{3.4}$$

Since $A = U\Lambda_A U^T$, we have

$$A - \lambda I_n = U(\begin{bmatrix} \lambda_1-\lambda & 0 & \dots & 0 \\ 0 & \lambda_2-\lambda & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n-\lambda \end{bmatrix})U^T = U(\Lambda_A - \lambda I_n)U^T$$

and, since $U$ is orthogonal,

$$\det(A - \lambda I_n) = \det(U(\Lambda_A - \lambda I_n)U^T) = \det(\Lambda_A - \lambda I_n) = \prod_{j=1}^{n}(\lambda_j - \lambda). \tag{3.5}$$

Then, as $(U\Lambda_A U^T)^{-1} = U(\Lambda_A)^{-1}U^T$,

$$(A - \lambda I_n)^{-1} = U(\begin{bmatrix} \frac{1}{\lambda_1-\lambda} & 0 & \dots & 0 \\ 0 & \frac{1}{\lambda_2-\lambda} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\lambda_n-\lambda} \end{bmatrix})U^T.$$

Plugging this into Schur's formula, we get

$$B^T(A - \lambda I_n)^{-1}B = B^T U(\begin{bmatrix} \frac{1}{\lambda_1-\lambda} & 0 & \dots & 0 \\ 0 & \frac{1}{\lambda_2-\lambda} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\lambda_n-\lambda} \end{bmatrix})U^T B = \sum_{j=1}^{n} \frac{1}{\lambda_j - \lambda}(\vec{u}_j{}^T B)(\vec{u}_j{}^T B)^T = \sum_{j=1}^{n} \frac{(\vec{u}_j{}^T B)^2}{\lambda_j - \lambda}. \tag{3.6}$$

Combining 3.5 and 3.6 in Equation 3.4 gives us

$$\det(D(T) - \lambda I_{n+m}) = \left(\prod_{j=1}^{n}(\lambda_j - \lambda)\right) \cdot \det\left(C - \lambda I_m - \sum_{j=1}^{n} \frac{(\vec{u}_j{}^T B)^2}{\lambda_j - \lambda}\right).$$

as desired. □

In this section, we have shown the conditions under which eigenvalues of submatrices re-appear in the spectrum of the full matrix, and how we can compute the remaining eigenvalues. In next section, we will discuss the eigenvalue of cherries.

## 3.2. Characterizing cherries via eigenvalues

As we have seen in Figures 3.1 and 3.2, $T_{17,1}$ and $T_{17,2}$ both have five cherries and eigenvalue $-2$ with multiplicity 5 in their spectra. Using Lemma 3, we will show that a cherry indeed always results in an eigenvalue $-2$.

**Lemma 5.** *Let $T$ be a tree on $n$ leaves and let $D(T)$ be its pairwise distance matrix. The tree contains $k$ cherries on the leaves $(l_{i_1}, l_{j_1}), \ldots, (l_{i_k}, l_{j_k})$ if and only if eigenvalue $-2$ has multiplicity $k$ with eigenvectors $\vec{v}_r = -e_{i_r} + e_{j_r}$ for $r = 1, \ldots, k$.*

*Proof.* We start with the reversed direction. Suppose that the eigenvalue $-2$ of $D(T)$ has multiplicity $k$ with corresponding eigenvectors $-e_{i_r} + e_{j_r}$ for $r = 1, ..., k$. Pick one $r$ arbitrarily and denote its eigenvector by $\vec{v}_r = \vec{v}$. Note that $v_i = -1$, $v_j = 1$ and $v_q = 0$ for $q \neq i, j$. By definition, we must have $D(T)\vec{v} = -2\vec{v}$. We will discuss the different rows separately. First, at row $i$, we have

$$\sum_{p=1}^{n} D_{ip} v_p = D_{ii} v_i + D_{ij} v_j = -2 v_i,$$

so we have

$$D_{ij} = 2.$$

Therefore, we must have $D_{ij} = 2$, which immediately implies $D_{ji} = 2$. For any $q \neq i, j$, we have

$$\sum_{p=1}^{n} D_{qp} v_p = D_{qi} v_i + D_{qj} v_j = 0,$$

which implies

$$D_{qi} = D_{qj}.$$

Therefore, since the distance between $l_i = l_{i_r}$ and $l_j = l_{j_r}$ is 2 and every third leaf is equidistant to $l_i$ and $l_j$, we have a cherry on leaves $(l_{i_r}, l_{j_r})$. Since eigenvalue $-2$ appears $k$ times in the spectrum and $r$ is an arbitrary integer between 1 and $k$, we can apply the above $k$ times. That results in $k$ cherries.

For the other direction, suppose $T$ contains $k \geq 1$ cherries on leaves $(l_{i_1}, l_{j_1}), \ldots, (l_{i_k}, l_{j_k})$. Note that there is at least one cherry, since the lowest interior vertex of any path within the tree must have two children. We first take a look at $k = 1$. Let $i = i_1$ be an arbitrary number between 1 and $n-1$ and let $j = j_1 = i + 1$. Then

$$D(T) = \begin{bmatrix} A & & B & & C \\ B^T & \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}_1 & & E \\ C^T & & E^T & & F \end{bmatrix}$$

where $\left[\begin{smallmatrix} 0 & 2 \\ 2 & 0 \end{smallmatrix}\right]_1$ stands for the matrix corresponding to the cherry on leaves $(l_i, l_j) = (l_{i_1}, l_{j_1})$. This matrix has eigenvalues $-2$ and 2 with eigenvectors $\left[\begin{smallmatrix} -1 \\ 1 \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right]$, respectively. We want to show that $-2$ is an eigenvalue of $D(T)$.

Observe that for any leaf $l_q$, where $q \neq i, j$, $d(l_q, l_i) = d(l_q, l_j)$. This means that

$$B = \begin{bmatrix} d(l_1, l_i) & d(l_1, l_j) \\ \vdots & \vdots \\ d(l_{i-1}, l_i) & d(l_{i-1}, l_j) \end{bmatrix} = \begin{bmatrix} d(l_1, l_i) & d(l_1, l_i) \\ \vdots & \vdots \\ d(l_{i-1}, l_i) & d(l_{i-1}, l_i) \end{bmatrix}$$

and

$$E^T = \begin{bmatrix} d(l_{i+2}, l_i) & d(l_{i+2}, l_j) \\ \vdots & \vdots \\ d(l_n, l_i) & d(l_n, l_j) \end{bmatrix} = \begin{bmatrix} d(l_{i+2}, l_i) & d(l_{i+2}, l_i) \\ \vdots & \vdots \\ d(l_n, l_i) & d(l_n, l_i) \end{bmatrix}.$$

In addition, observe that, for the eigenvector $\left[\begin{smallmatrix} -1 \\ 1 \end{smallmatrix}\right]$ of eigenvalue $-2$,

$$B \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} d(l_1, l_i) & d(l_1, l_i) \\ \vdots & \vdots \\ d(l_{i-1}, l_i) & d(l_{i-1}, l_i) \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -d(l_1, l_i) + d(l_1, l_i) \\ \vdots \\ -d(l_{i-1}, l_i) + d(l_{i-1}, l_i) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

and

$$E^T \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} d(l_{i+2}, l_i) & d(l_{i+2}, l_i) \\ \vdots & \vdots \\ d(l_n, l_i) & d(l_n, l_i) \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -d(l_{i+2}, l_i) + d(l_{i+2}, l_i) \\ \vdots \\ -d(l_n, l_i) + d(l_n, l_i) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Now, by Lemma 3, since $-2$ is an eigenvalue of $D(T') = \left[\begin{smallmatrix} 0 & 2 \\ 2 & 0 \end{smallmatrix}\right]_1$ and $B\left[\begin{smallmatrix} -1 \\ 1 \end{smallmatrix}\right] = \vec{0}$ and $E^T\left[\begin{smallmatrix} -1 \\ 1 \end{smallmatrix}\right] = \vec{0}$, we know that $-2$ is an eigenvalue of $D(T)$ with eigenvector $\begin{bmatrix} \vec{0} \\ -1 \\ 1 \\ \vec{0} \end{bmatrix} = -e_i + e_j = -e_{i_1} + e_{j_1}$, as desired. Now, for $k$ cherries, we can label the leaves of each cherry by $i$ and $j = i+1$, where $i = 1, 3, \ldots, 2k-1$. Therefore, for $k$ cherries, we must have

$$D(T) = \begin{bmatrix} \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}_1 & \cdots & \cdots & \cdots & \\ \vdots & \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}_2 & \cdots & \cdots & \\ \vdots & \vdots & \ddots & & \\ \vdots & \vdots & & \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}_k & \\ & & & & C \end{bmatrix}$$

Now we can apply Lemma 3 $k$ times, in the same way as above. Therefore, every cherry $(l_{i_r}, l_{j_r})$, where $r = 1, \ldots, k$, leads to eigenvalue $-2$ with corresponding eigenvector $-e_{i_r} + e_{j_r}$.

□

In next section, we will show that not only cherries lead to specific eigenvalues: perfectly balanced trees of an arbitrary height $h$ have a closed-form spectrum as well.

## 3.3. Spectra of perfectly balanced trees

In this section, we want to apply Lemma 3 to trees $T$ with a perfectly balanced pendant subtree $T'$. It turns out that all negative eigenvalues of $D(T')$ are eigenvalues of $D(T)$ as well, which we will show in Lemma 8. In addition, these perfectly balanced (sub)trees always lead so specific eigenvalues, which we will show in Theorem 2. Before we are able to prove these results, we need a few lemmas and corollaries, which we will state and prove here.

**Lemma 6.** *Every pairwise distance matrix $D(T)$ on $m$ leaves has exactly one positive eigenvalue.*

*Proof.* We know that the pairwise distance matrix $D(T)$ on $m$ leaves is a principal submatrix of the distance matrix on the full set of vertices, say $A(T)$ on $n > m$ vertices. In other words, we obtain $D(T)$ by removing the rows and columns from $A(T)$ that correspond to vertices other than leaves. Denote the eigenvalues of $A(T)$ by $\lambda$ and those of $D(T)$ by $\mu$. Since $\sum_{i=1}^{m} \mu_i = 0$ by Corollary 1, and by Lemma 5 there is a negative eigenvalue, there must be at least one positive eigenvalue. Graham, Pollak and Merris already discovered that $A(T)$ has exactly one positive eigenvalue, say $\lambda_n$ [5, 11]. By Theorem 1, we have that $\lambda_m \leq \mu_m \leq \lambda_n$. Since $\lambda_m \leq \lambda_{n-1} < 0$, we must have $\mu_j < 0$ for all $j < m$. Thus $D(T)$ has at most one positive eigenvalue. In conclusion, $D(T)$ has exactly one positive eigenvalue. □

In next lemmas and corollaries, we will frequently use the following observations:

**Observation 1.** The perfectly balanced tree of height 1 is displayed in Figure 3.3. This tree has pairwise distance matrix $D(T) = \left[\begin{smallmatrix} 0 & 2 \\ 2 & 0 \end{smallmatrix}\right]$ with eigenvalues $-2$ and 2 and corresponding eigenvectors $\left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} -1 \\ 1 \end{smallmatrix}\right]$.

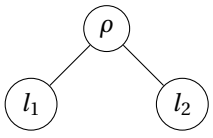



Figure 3.3: Perfectly balanced tree of height 1.

**Observation 2.** One can create the perfectly balanced tree $T_h$ by attaching two copies of $T_{h-1}$ to one root $\rho$, as displayed in Figure 3.4.

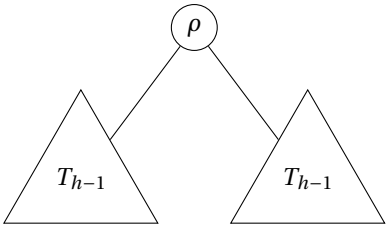



Figure 3.4: Perfectly balanced tree of height $h$, where two copies of $T_{h-1}$ are attached to one root $\rho$.

**Lemma 7.** *Let $T_h$ be a perfectly balanced tree of height $h$ with corresponding pairwise distance matrix $D(T_h)$. Then, for any $i$, $\sum_{j=1}^{2^h} D_{ij}(T_h) = \sum_{k=1}^{h} 2^k k$, where $D_{ij} = d(l_i, l_j)$.*

*Proof.* Let $T_h$ be a perfectly balanced tree of height $h$, with $n = 2^h$ leaves. Note that in a perfectly balanced tree, every leaf $l_i$ has the same set of distances to all other leaves. So, without loss of generality, we look at the sum of the entries in row $i$. We will prove the sum by induction on the height $h$. For the base case, pick a perfectly balanced tree of height 1, which is displayed in Figure 3.3. By Observation 1, the corresponding distance matrix is given by $D(T_1) = \left[\begin{smallmatrix} 0 & 2 \\ 2 & 0 \end{smallmatrix}\right]$. Here, $\sum_{j=1}^{2^1} D_{ij}(T_1) = 0+2 = 2^1 \cdot 1$. Now, for the induction hypothesis, assume that the perfectly balanced tree $T_{h-1}$ of height $h-1$ satisfies $\sum_{j=1}^{2^{h-1}} D_{ij}(T_{h-1}) = \sum_{k=1}^{h-1} 2^k k$. Now we want to show that $\sum_{j=1}^{2^h} D_{ij}(T_h) = \sum_{k=1}^{h} 2^k k$ holds as well. By Observation 2, we can create the tree $T_h$ by taking two copies of $T_{h-1}$ and gluing their roots onto a new root. Let the leaves in the left subtree be labeled from 1 to $2^{h-1}$ and the leaves in the right subtree from $2^{h-1}+1$ to $2^h$. Since every leaf in the subtree $T_{h-1}$ has distance $h$ to the root, we must have that for any leaf $l_i$ in the left subtree and any leaf $l_j$ in the right subtree, $D_{ij} = d(l_i, l_j) = 2h$. Therefore, the pairwise distance matrix corresponding to $T_h$ is given by:

$$D(T_h) = \begin{bmatrix} D(T_{h-1}) & 2hJ \\ 2hJ & D(T_{h-1}) \end{bmatrix}$$

where $J$ is the $2^{h-1} \times 2^{h-1}$ all-ones matrix. Note that

$$\sum_{j=1}^{2^h} D_{ij}(T_h) = \sum_{j=1}^{2^{h-1}} D_{ij}(T_{h-1}) + \sum_{j=1}^{2^{h-1}} 2h = \sum_{k=1}^{h-1} 2^k k + 2^{h-1} 2h = \sum_{k=1}^{h-1} 2^k k + 2^h h = \sum_{k=1}^{h} 2^k k$$

by the induction hypothesis. Hence, $\sum_{j=1}^{2^h} D_{ij}(T_h) = \sum_{k=1}^{h} 2^k k$ for any perfectly balanced tree of height $h$. □

Using Lemmas 6 and 7, we can prove the following:

**Corollary 2.** *For a perfectly balanced tree $T_h$ of height $h$, the largest eigenvalue has the all-ones vector $\vec{1}$ as eigenvector. For all other eigenvalues with corresponding eigenvector $\vec{u} = [u_1 \cdots u_n]^T$, we have that $\sum_{i=1}^{n} u_i = 0$.*

*Proof.* Let $T_h$ be a perfectly balanced tree of height $h$ with $n = 2^h$ leaves. In Lemma 7, we already saw that every row $i$ in $D(T_h)$ must have the same sum, $\sum_{j=1}^{n} D_{ij}(T_h) = \sum_{k=1}^{h} 2^k k$. This means that $D(T_h)\vec{1} = \left(\sum_{k=1}^{h} 2^k k\right)\vec{1}$. Hence, $\vec{1}$ is the eigenvector corresponding to eigenvalue $\lambda = \sum_{k=1}^{h} 2^k k$. By Lemma 6, there is only one positive eigenvalue, and since $\sum_{k=1}^{h} 2^k k > 0$, we must have that this is the largest eigenvalue. For the second part of the lemma, we use the fact that a symmetric matrix has orthogonal eigenvectors. That means that $\vec{u}^T \vec{v} = 0$ for two distinct eigenvectors $\vec{u}$ and $\vec{v}$. Thus, for $\vec{v} = \vec{1}$, $\vec{u}^T \vec{1} = \sum_{i=1}^{n} u_i = 0$, as desired. □

Now that we have seen these lemmas, we are able to prove the following result, which can be seen as a generalization of Lemma 5.

**Lemma 8.** *Let $T$ be a tree on $n$ leaves with a perfectly balanced pendant subtree $T_h$ on $m = 2^h$ leaves. Then all negative eigenvalues $\lambda$ of $D(T_h)$ with multiplicity $p$ and corresponding eigenvectors $\vec{u}$ are eigenvalues of $D(T)$ with multiplicity $q \geq p$ and corresponding eigenvectors $\begin{bmatrix} \vec{u} \\ \vec{0} \end{bmatrix}$.*

*Proof.* Let $T$ be a tree on $n$ leaves with a perfectly balanced pendant subtree $T_h$ on $m = 2^h$ leaves, as displayed in Figure 3.5. $T'$ is just the remaining part of the tree.

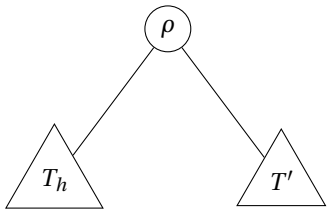



Figure 3.5: A tree $T$ with a perfectly balanced pendant subtree $T_h$.

The corresponding distance matrix is given by:

$$D(T) = \begin{bmatrix} D(T_h) & C \\ C^T & D(T') \end{bmatrix}$$

Note that all leaves in the subtree $T_h$ are equidistant to all other leaves in the tree, which means that the rows in $C$ (or equivalently, the columns in $C^T$) are equal. For any negative eigenvalue $\lambda$ of $D(T_h)$ with corresponding eigenvector $\vec{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}$:

$$C^T \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} = \begin{bmatrix} c_1 & \dots & c_1 \\ \vdots & & \vdots \\ c_{n-m} & \dots & c_{n-m} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} = \begin{bmatrix} c_1 \sum_{i=1}^m u_i \\ \vdots \\ c_{n-m} \sum_{i=1}^m u_i \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

where we used Corollary 2 for the last equality. Then, by Lemma 3, we have that $\lambda$ is an eigenvalue of $D(T)$ with corresponding eigenvector $\begin{bmatrix} \vec{u} \\ \vec{0} \end{bmatrix}$. Since this holds for all negative eigenvalues $\lambda$ of $D(T_h)$, we must have that $\lambda$ is an eigenvalue of $D(T)$ with multiplicity $q \geq p$. Note that if $D(T')$ itself contains a perfectly balanced pendant subtree with the same eigenvalue $\lambda$, $q > p$. □

Using the above lemmas, we can show that perfectly balanced trees only have negative eigenvalues of the form $-2(2^k - 1)$, where $k$ is a number between 1 and the height $h$ of the tree. As already mentioned in Chapter 1, Singh et al. stated that these are the only negative eigenvalues [14]. In the following theorem, we will also prove this result.

**Theorem 2.** *For a perfectly balanced tree of height h, the negative eigenvalues are of the form* $-2(2^k - 1)$ *with multiplicity* $2^{h-k}$ *for* $k = 1, \dots, h$. *The most negative eigenvalue* $-2(2^h - 1)$ *has eigenvector* $\begin{bmatrix} -\vec{1} \\ \vec{1} \end{bmatrix}$.

*Proof.* We will prove this theorem by induction on the height $h$. For the base case of height 1, we have the perfectly balanced tree $T_1$, which is displayed in Figure 3.3. By Observation 1, the corresponding distance matrix $D(T_1)$ has eigenvalues 2 and $-2$ with corresponding eigenvectors $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$. Thus, indeed, the negative eigenvalue equals $-2(2^1 - 1) = -2$ with multiplicity $2^{1-1} = 1$ and corresponding eigenvector $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$. For the induction hypothesis, we assume that $T_{h-1}$ has negative eigenvalues $-2(2^k - 1)$ with multiplicity $2^{h-1-k}$ for $k = 1, \dots, h-1$. Moreover, we assume that $\begin{bmatrix} -\vec{1} \\ \vec{1} \end{bmatrix}$ is the eigenvector corresponding to $\lambda = -2(2^{h-1} - 1)$. It suffices to show that $T_h$ has negative eigenvalues $-2(2^k - 1)$ with multiplicity $2^{h-k}$ for $k = 1, .., h$ and $\begin{bmatrix} -\vec{1} \\ \vec{1} \end{bmatrix}$ is the eigenvector corresponding to $\lambda = -2(2^h - 1)$. We again use Observation 2. Note that the distance matrix corresponding to $T_h$ is given by:

$$D(T_h) = \begin{bmatrix} D(T_{h-1}) & 2hJ \\ 2hJ & D(T_{h-1}) \end{bmatrix}$$

where $J$ is the $2^{h-1} \times 2^{h-1}$ all-ones matrix. Note that by Lemmas 3 and 8, we must have that all negative eigenvalues $\lambda_1, \dots, \lambda_{h-2}$ of $D(T_{h-1})$ are eigenvalues of $D(T_h)$ as well with twice the multiplicity and corresponding eigenvectors $\begin{bmatrix} \vec{u}_i \\ \vec{0} \end{bmatrix}$ and $\begin{bmatrix} \vec{0} \\ \vec{u}_i \end{bmatrix}$ for $\lambda_i$. That means that $-2(2^k - 1)$ are eigenvalues with multiplicity $2 \cdot 2^{h-1-k} = 2^{h-k}$ for $k = 1, \dots, h-1$. Now it remains to show that $-2(2^h - 1)$ is an eigenvalue with multiplicity

$2^{h-h} = 1$ as well. Suppose that $\begin{bmatrix} \vec{-1} \\ \vec{1} \end{bmatrix}$ is the eigenvector corresponding to the remaining eigenvalue. Denote the remaining eigenvalue by $\Lambda$. Now, we must solve $D(T_h)\begin{bmatrix} \vec{-1} \\ \vec{1} \end{bmatrix} = \Lambda \begin{bmatrix} \vec{-1} \\ \vec{1} \end{bmatrix}$. Thus, using Lemma 7, and using the fact that $\sum_{i=0}^{n-1} i a^i = \frac{a - n a^n + (n-1)a^{n+1}}{(1-a)^2}$, we see that:

$$D(T_h)\begin{bmatrix} \vec{-1} \\ \vec{1} \end{bmatrix} = \begin{bmatrix} D(T_{h-1}) & 2hJ \\ 2hJ & D(T_{h-1}) \end{bmatrix}\begin{bmatrix} \vec{-1} \\ \vec{1} \end{bmatrix} = \begin{bmatrix} -\sum_{k=1}^{h-1} 2^k k + 2^{h-1}2h \\ -2^{h-1}2h + \sum_{k=1}^{h-1} 2^k k \end{bmatrix} =$$
$$\begin{bmatrix} -(2 - h2^h + (h-1)2^{h+1}) + 2^{h-1}2h \\ -2^{h-1}2h + (2 - h2^h + (h-1)2^{h+1}) \end{bmatrix} = \begin{bmatrix} 2(2^h - 1) \\ -2(2^h - 1) \end{bmatrix} = -2(2^h - 1)\begin{bmatrix} \vec{-1} \\ \vec{1} \end{bmatrix}$$

which indeed states that $-2(2^h - 1)$ is an eigenvalue of $D(T_h)$ with eigenvector $\begin{bmatrix} \vec{-1} \\ \vec{1} \end{bmatrix}$. □

## 3.4. Explanation of spectral equivalence in distinct trees

Now we want to apply the results of the previous sections to the phylogenetic trees $T_{17,1}$ and $T_{17,2}$, to see if we can explain why they share a spectrum. Note that the pairwise distance matrix of a cherry, $\left[\begin{smallmatrix} 0 & 2 \\ 2 & 0 \end{smallmatrix}\right]$, appears five times as a submatrix in both $D(T_{17,1})$ and $D(T_{17,2})$. In addition, $D(T_2) = \left[\begin{smallmatrix} 0 & 2 & 4 & 4 \\ 2 & 0 & 4 & 4 \\ 4 & 4 & 0 & 2 \\ 4 & 4 & 2 & 0 \end{smallmatrix}\right]$ appears once. By Lemma 8 and Theorem 2, we see that $D(T_{17,1})$ indeed must have $(-6)^1$ and $(-2)^5$ in its spectrum.

For the remaining eigenvalues, we will use Lemma 3 and Lemma 4. Let $A$ be the $16 \times 16$ top-left matrix that appears in both $D(T_{17,1})$ and $D(T_{17,2})$. Moreover, let $\vec{b_1} = [8,8,8,8,7,7,5,5,5,5,6,6,5,3,4,4]^T$ and $\vec{b_2} = [5,5,5,5,6,6,8,8,8,8,3,5,4,4,7,7]^T$. Then $D(T_{17,1}) = \begin{bmatrix} A & \vec{b_1} \\ \vec{b_1}^T & 0 \end{bmatrix}$ and $D(T_{17,2}) = \begin{bmatrix} A & \vec{b_2} \\ \vec{b_2}^T & 0 \end{bmatrix}$. In addition to eigenvalues $-2$ and $-6$, $-11.123$ and $-2.877$ appear in the spectra of $A$, $D(T_{17,1})$ and $D(T_{17,2})$. The Python code in Appendix A verifies that the eigenvectors corresponding to these eigenvalues are indeed orthogonal to $\vec{b_1}$ and $\vec{b_2}$. Then, Lemma 3 explains why the eigenvalues are inherited.

For the remaining eigenvalues of $D(T_{17,1})$ and $D(T_{17,2})$, we will use Lemma 4. The same Python code verifies that for all unit eigenvectors $\vec{u}$ that are not orthogonal to $\vec{b_1}$ and $\vec{b_2}$, $(\vec{u}^T\vec{b_1})^2 = (\vec{u}^T\vec{b_2})^2$. That means that

$$\det(D(T_{17,1}) - \lambda I) = \left(\prod_{j=1}^{16}(\lambda_j - \lambda)\right) \cdot \det\left(-\lambda - \sum_{j=1}^{16} \frac{(\vec{u_j}^T\vec{b_1})^2}{\lambda_j - \lambda}\right) =$$
$$\left(\prod_{j=1}^{16}(\lambda_j - \lambda)\right) \cdot \det\left(-\lambda - \sum_{j=1}^{16} \frac{(\vec{u_j}^T\vec{b_2})^2}{\lambda_j - \lambda}\right) = \det(D(T_{17,2}) - \lambda I).$$

The above explains why $D(T_{17,1})$ and $D(T_{17,2})$ have the exact same spectrum.

## 3.5. Other interesting eigenvalues

In Section 3.3, we showed that negative integer eigenvalues of perfectly balanced trees are restricted to the form $-2(2^k - 1)$. In this section, we want to discover whether these are the only possible negative integer eigenvalues in all trees. Moreover, we will investigate other eigenvalues that might reveal the structure of a (sub)tree.

For small examples, the closed-form on the negative eigenvalues discussed in th Section 3.3 seemed to be the only negative integer eigenvalues possible. However, using Python we found that the negative integer eigenvalues of general pairwise distance matrices are not restricted to this form. In Appendix B, we added a Python code that generates pairwise distance matrices of trees up to $n$ leaves. Using this code, we found that there exist pairwise distance matrices with other negative integer eigenvalues. One example is the tree in Figure 3.6, which has eigenvalue $-4$ as well. This means that negative integer eigenvalues of pairwise distance matrices are not restricted to the form $-2(2^k - 1)$, where $k$ is a number between 1 and the height of the tree.

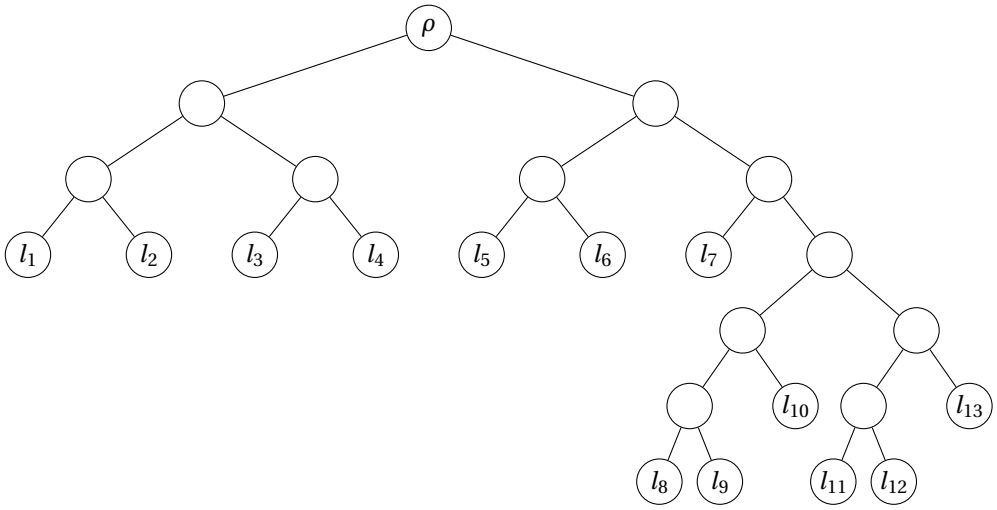



Figure 3.6: A phylogenetic tree $T'$ on 13 leaves with $-4$ in its spectrum.

The following matrix is the pairwise distance matrix corresponding to the tree in Figure 3.6:

$$D(T') = \begin{bmatrix} 0 & 2 & 4 & 4 & 6 & 6 & 6 & 9 & 9 & 8 & 9 & 9 & 8 \\ 2 & 0 & 4 & 4 & 6 & 6 & 6 & 9 & 9 & 8 & 9 & 9 & 8 \\ 4 & 4 & 0 & 2 & 6 & 6 & 6 & 9 & 9 & 8 & 9 & 9 & 8 \\ 4 & 4 & 2 & 0 & 6 & 6 & 6 & 9 & 9 & 8 & 9 & 9 & 8 \\ 6 & 6 & 6 & 6 & 0 & 2 & 4 & 7 & 7 & 6 & 7 & 7 & 6 \\ 6 & 6 & 6 & 6 & 2 & 0 & 4 & 7 & 7 & 6 & 7 & 7 & 6 \\ 6 & 6 & 6 & 6 & 4 & 4 & 0 & 5 & 5 & 4 & 5 & 5 & 4 \\ 9 & 9 & 9 & 9 & 7 & 7 & 5 & 0 & 2 & 3 & 6 & 6 & 5 \\ 9 & 9 & 9 & 9 & 7 & 7 & 5 & 2 & 0 & 3 & 6 & 6 & 5 \\ 8 & 8 & 8 & 8 & 6 & 6 & 4 & 3 & 3 & 0 & 5 & 5 & 4 \\ 9 & 9 & 9 & 9 & 7 & 7 & 5 & 6 & 6 & 5 & 0 & 2 & 3 \\ 9 & 9 & 9 & 9 & 7 & 7 & 5 & 6 & 6 & 5 & 2 & 0 & 3 \\ 8 & 8 & 8 & 8 & 6 & 6 & 4 & 5 & 5 & 4 & 3 & 3 & 0 \end{bmatrix},$$

which has spectrum $\{(-27.616)^1, (-11.123)^1, (-9.975)^1, (-6)^1, (-4)^1, (-3.006)^1, (-2.877)^1, (-2)^5, (74.598)^1\}$, where each non-integer eigenvalue is rounded to three decimals.

In this spectrum, we see something interesting. Note that the eigenvalues $-11.123$ and $-2.877$ also appear in the spectra of $D(T_{17,1})$ and $D(T_{17,2})$. For the matrix $D(T')$, the eigenvectors corresponding to eigenvalues $-11.123$ and $-2.877$ are

$$[0,0,0,0,0,0,0,-0.465,-0.465,-0.261,0.465,0.465,0.261]^T$$

$$[0,0,0,0,0,0,0,0.185,0.185,-0.657,-0.185,-0.185,0.657]^T,$$

respectively. Note that the non-zero entries appear at indices $8,9,10,11,12,13$. If we look at the leaves in Figure 3.6 with these indices as labels, we see a pendant subtree on leaves $\{l_8, l_9, l_{10}, l_{11}, l_{12}, l_{13}\}$.

For the matrices $D(T_{17,1})$ and $D(T_{17,2})$, the eigenvectors corresponding to eigenvalues $-11.123$ and $-2.877$ are

$$[0,0,0,0,0,0,-0.465,-0.465,0.465,0.465,0,0,0,0,-0.261,0.261,0]^T,$$

$$[0,0,0,0,0,0,0.185,0.185,-0.185,-0.185,0,0,0,0,-0.657,0.657,0]^T,$$

respectively. The non-zero entries appear at indices $7,8,9,10,15$ and $16$. If we look at the leaves in $T_{17,1}$ and $T_{17,2}$ with these indices as labels, we see the same pendant subtree on leaves $\{l_7, l_8, l_9, l_{10}, l_{15}, l_{16}\}$. Therefore, these eigenvalues might indicate the presence of subtrees of that form. This is something that remains to be (dis)proven.

# 4

# Conclusion and discussion

In this thesis, we investigated how spectra of pairwise distance matrices reveal structural information about rooted binary phylogenetic trees. Although the spectrum does not uniquely determine the full topology of the tree, as we saw for the two distinct trees $T_{17,1}$ and $T_{17,2}$, we showed that some eigenvalues can be highly informative.

In particular, in Section 3.2, we discovered that eigenvalue $-2$ with multiplicity $k$ in the spectrum, with associated eigenvector of the form $-e_i + e_j$, reveals the presence of $k$ cherries in the tree. In all rooted binary phylogenetic trees we have taken a look at, $-2$ always corresponds to a cherry. Therefore, we conjecture that the presence of an eigenvalue $-2$ is sufficient to guarantee the presence of a cherry, regardless of its eigenvector.

**Conjecture 1.** Let $T$ be a tree on $n$ leaves. Then the multiplicity of eigenvalue $-2$ in the spectrum of its pairwise distance matrix equals the number of cherries in the tree.

To prove this Conjecture, we must verify that for a general pairwise distance matrix $D(T)$, $D(T)\vec{v} = -2\vec{v}$, or similarly, $\sum_{j=1}^{n} d(l_i, l_j)v_j = -2v_i$, must imply that $d(l_i, l_j) = 2$ for some $i, j$. Note that $-2$ with multiplicity $k$ means there are $k$ different pairs of leaves with up-down distance 2. Another possibility is to prove that eigenvectors corresponding to eigenvalue $-2$ can only be of the form $-e_i + e_j$; then we can leave out the assumption that $\vec{v_r} = -e_{i_r} + e_{j_r}$ for cherry $r$ in Lemma 5.

In Section 3.1, we showed that an eigenvalue $\lambda$ of a submatrix (pendant subtree) can also appear as an eigenvalue of the full matrix, but only under a specific condition. According to Lemma 3, this occurs when the matrix connecting the subtree to the rest of the tree is orthogonal to an eigenvector of the submatrix corresponding to $\lambda$. In Lemma 8, we saw that the negative eigenvalues of a perfectly balanced pendant subtree with eigenvector $\vec{u}$ always re-appear in the spectrum of the entire tree with eigenvector $\begin{bmatrix} \vec{u} \\ \vec{0} \end{bmatrix}$. In Theorem 2, we proved that the negative eigenvalues of perfectly balanced trees are of the closed-form $-2(2^k - 1)$. We conjecture that all eigenvalues of the closed-form $-2(2^k - 1)$ reveal the presence of a perfectly balanced subtree of height $k$, regardless of their eigenvectors.

**Conjecture 2.** Let $T$ be a tree on $n$ leaves. Then the multiplicity of eigenvalue $-2(2^k - 1)$ in the spectrum of its pairwise distance matrix equals the number of perfectly balanced subtrees of height $k$ in the tree.

To prove this, a similar method to the proof of Conjecture 1 might be used.

Furthermore, in Section 3.5, we saw that that eigenvalues $-11.123$ and $2.877$ appeared in different trees with the same pendant subtree, which is displayed in Figure 4.1. We found that, for these trees, the labels of the leaves matched with the non-zero entries in the eigenvectors corresponding to $-11.123$ and $2.877$. These eigenvalues might indicate the presence of a pendant subtree of the form in Figure 4.1. Since we only looked at three trees with such a pendant subtree, it might be interesting to investigate whether this holds in general.

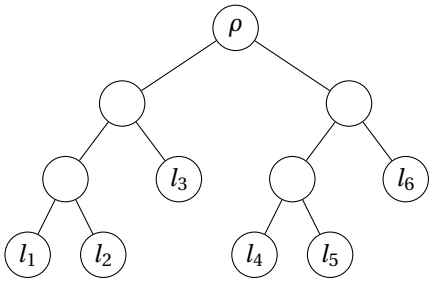



Figure 4.1: A pendant subtree that might corresponds to eigenvalues −11.123 and 2.877.

Besides this example, there might be other eigenvalues that reveal common subtree types, such as caterpillar trees. Another area for future research is non-binary or unrooted trees. In addition, we only considered unit edge length, while in reality, weighted edges are possible as well. This could be taken into account in future work. Lastly, spectra of phylogenetic networks could be investigated. Phylogenetic networks allow reticulations: vertices with two parents and one child. Therefore, multiple up-down paths are possible between pairs of leaves, resulting in a multiset matrix $D(T)$. It might be interesting to look at the shortest distance in such multiset-matrices. Defining spectra for such multiset-matrices and investigating what those spectra reveal about the topology could be very valuable in phylogenetics, as phylogenetic networks provide a more accurate representation of evolution than phylogenetic trees [2].

# Bibliography

The following articles and books were used for writing this thesis. In addition, we sometimes used ChatGPT as a search engine and as a tool for the writing part.


[1] Blagojevic, F. & Nikolopoulos, D. & Stamatakis, A. & Antonopoulos, C.D.. (2007). Dynamic Multigrain Parallelization on the Cell Broadband Engine. 90-100. 10.1145/1229428.1229445.

[2] Bordewich, M. & Semple, C.. (2016). Determining phylogenetic networks from inter-taxa distances. doi: 10.1007/s00285-015-0950-8. Epub 2015 Dec 14. PMID: 26666756.

[3] Fraleigh, J.B. & Beauregard, R.A.. (1995). Linear Algebra, 3rd edition. University of the District of Columbia.

[4] Graham, R.L. & Lovász, L.. (1978). Distance Matrix Polynomials of Trees. JATE Bolyai Intézet, Szeged, Hungary.

[5] Graham, R.L. & Pollak, H.O.. (1971). On the addressing problem for loop switching (1971). Bell System Tech. J., Vol. 50, 1971, pp. 2495-259.

[6] Hillis, D.M.. (1997). Phylogenetic analysis. doi: 10.1016/s0960-9822(97)70070-8. PMID: 9162471.

[7] Horn, R.A. & Johnson, C.R.. (2013). Matrix Analysis, 2nd edition. Cambridge University Press. ISBN 978-0-521-83940-2

[8] Janzen, T., Etienne, R.S.. (2024). Phylogenetic tree statistics: A systematic overview using the new R package 'treestats'. doi: 10.1016/j.ympev.2024.108168. Epub 2024 Aug 6. PMID: 39117295.

[9] Li, T., Liu, D., Yang, Y. et al. Phylogenetic supertree reveals detailed evolution of SARS-CoV-2. Sci Rep 10, 22366 (2020). https://doi.org/10.1038/s41598-020-79484-8

[10] Matsen, F.A., Evans, S.N.. (2012). Ubiquity of synonymity: almost all large binary trees are not uniquely identified by their spectra or their immanantal polynomials. https://doi.org/10.1186/1748-7188-7-14

[11] Merris, R.. (1990). The Distance Spectrum of a Tree. CCC 0364-9024/90/030365-05$04.OO

[12] De Ponte, M.A. & de Campos, L.C.. (2018). Determining the eigenvalues of a square matrix through known information of its submatrix. 10.48550/arXiv.1810.10087.

[13] Semple, C., & Steel, M.. (2003). Phylogenetics (Vol. 24). Oxford University Press on Demand.

[14] Singh, A., Krishnamurthy, A., Balakrishnan, S. & Xu, M.. (2012). Completion of high-rank ultrametric matrices using selective entries. https://doi.org/10.1109/SPCOM.2012.6290247

[15] Vézquez, D.P. & Gittleman, J.L.. (2004). Biodiversity conservation: Does phylogeny matter? https://doi.org/10.1016/S0960-9822(98)70242-8

[16] Wilson, R.J.. (1996). Introduction to Graph Theory, Fourth Edition. ISBN 0-582-24993-7

# A

# Python code for computing spectra

```
    import numpy as np

''' Pairwise distance matrix corresponding to T_{17,1}, which does not have a unique spectrum '''
D_17_1 =  [[ 0, 2, 4, 4, 9, 9,11,11,11,11, 4, 8, 7, 7,10,10, 8],
           [ 2, 0, 4, 4, 9, 9,11,11,11,11, 4, 8, 7, 7,10,10, 8],
           [ 4, 4, 0, 2, 9, 9,11,11,11,11, 4, 8, 7, 7,10,10, 8],
           [ 4, 4, 2, 0, 9, 9,11,11,11,11, 4, 8, 7, 7,10,10, 8],
           [ 9, 9, 9, 9, 0, 2,10,10,10,10, 7, 3, 4, 6, 9, 9, 7],
           [ 9, 9, 9, 9, 2, 0,10,10,10,10, 7, 3, 4, 6, 9, 9, 7],
           [11,11,11,11,10,10, 0, 2, 6, 6, 9, 9, 8, 6, 3, 5, 5],
           [11,11,11,11,10,10, 2, 0, 6, 6, 9, 9, 8, 6, 3, 5, 5],
           [11,11,11,11,10,10, 6, 6, 0, 2, 9, 9, 8, 6, 5, 3, 5],
           [11,11,11,11,10,10, 6, 6, 2, 0, 9, 9, 8, 6, 5, 3, 5],
           [ 4, 4, 4, 4, 7, 7, 9, 9, 9, 9, 0, 6, 5, 5, 8, 8, 6],
           [ 8, 8, 8, 8, 3, 3, 9, 9, 9, 9, 6, 0, 3, 5, 8, 8, 6],
           [ 7, 7, 7, 7, 4, 4, 8, 8, 8, 8, 5, 3, 0, 4, 7, 7, 5],
           [ 7, 7, 7, 7, 6, 6, 6, 6, 6, 6, 5, 5, 4, 0, 5, 5, 3],
           [10,10,10,10, 9, 9, 3, 3, 5, 5, 8, 8, 7, 5, 0, 4, 4],
           [10,10,10,10, 9, 9, 5, 5, 3, 3, 8, 8, 7, 5, 4, 0, 4],
           [ 8, 8, 8, 8, 7, 7, 5, 5, 5, 5, 6, 6, 5, 3, 4, 4, 0]]

''' Pairwise distance matrix corresponding to T_{17,2}, which does not have a unique spectrum '''
D_17_2 = [[ 0, 2, 4, 4, 9, 9,11,11,11,11, 4, 8, 7, 7,10,10, 5],
          [ 2, 0, 4, 4, 9, 9,11,11,11,11, 4, 8, 7, 7,10,10, 5],
          [ 4, 4, 0, 2, 9, 9,11,11,11,11, 4, 8, 7, 7,10,10, 5],
          [ 4, 4, 2, 0, 9, 9,11,11,11,11, 4, 8, 7, 7,10,10, 5],
          [ 9, 9, 9, 9, 0, 2,10,10,10,10, 7, 3, 4, 6, 9, 9, 6],
          [ 9, 9, 9, 9, 2, 0,10,10,10,10, 7, 3, 4, 6, 9, 9, 6],
          [11,11,11,11,10,10, 0, 2, 6, 6, 9, 9, 8, 6, 3, 5, 8],
          [11,11,11,11,10,10, 2, 0, 6, 6, 9, 9, 8, 6, 3, 5, 8],
          [11,11,11,11,10,10, 6, 6, 0, 2, 9, 9, 8, 6, 5, 3, 8],
          [11,11,11,11,10,10, 6, 6, 2, 0, 9, 9, 8, 6, 5, 3, 8],
          [ 4, 4, 4, 4, 7, 7, 9, 9, 9, 9, 0, 6, 5, 5, 8, 8, 3],
          [ 8, 8, 8, 8, 3, 3, 9, 9, 9, 9, 6, 0, 3, 5, 8, 8, 5],
          [ 7, 7, 7, 7, 4, 4, 8, 8, 8, 8, 5, 3, 0, 4, 7, 7, 4],
          [ 7, 7, 7, 7, 6, 6, 6, 6, 6, 6, 5, 5, 4, 0, 5, 5, 4],
          [10,10,10,10, 9, 9, 3, 3, 5, 5, 8, 8, 7, 5, 0, 4, 7],
          [10,10,10,10, 9, 9, 5, 5, 3, 3, 8, 8, 7, 5, 4, 0, 7],
          [ 5, 5, 5, 5, 6, 6, 8, 8, 8, 8, 3, 5, 4, 4, 7, 7, 0]]
```

```
''' Top-left 16x16 matrix that is a submatrix of both D_17_1 and D_17_2 '''
D_16 =   [[ 0, 2, 4, 4, 9, 9,11,11,11,11, 4, 8, 7, 7,10,10],
          [ 2, 0, 4, 4, 9, 9,11,11,11,11, 4, 8, 7, 7,10,10],
          [ 4, 4, 0, 2, 9, 9,11,11,11,11, 4, 8, 7, 7,10,10],
          [ 4, 4, 2, 0, 9, 9,11,11,11,11, 4, 8, 7, 7,10,10],
          [ 9, 9, 9, 9, 0, 2,10,10,10,10, 7, 3, 4, 6, 9, 9],
          [ 9, 9, 9, 9, 2, 0,10,10,10,10, 7, 3, 4, 6, 9, 9],
          [11,11,11,11,10,10, 0, 2, 6, 6, 9, 9, 8, 6, 3, 5],
          [11,11,11,11,10,10, 2, 0, 6, 6, 9, 9, 8, 6, 3, 5],
          [11,11,11,11,10,10, 6, 6, 0, 2, 9, 9, 8, 6, 5, 3],
          [11,11,11,11,10,10, 6, 6, 2, 0, 9, 9, 8, 6, 5, 3],
          [ 4, 4, 4, 4, 7, 7, 9, 9, 9, 9, 0, 6, 5, 5, 8, 8],
          [ 8, 8, 8, 8, 3, 3, 9, 9, 9, 9, 6, 0, 3, 5, 8, 8],
          [ 7, 7, 7, 7, 4, 4, 8, 8, 8, 8, 5, 3, 0, 4, 7, 7],
          [ 7, 7, 7, 7, 6, 6, 6, 6, 6, 6, 5, 5, 4, 0, 5, 5],
          [10,10,10,10, 9, 9, 3, 3, 5, 5, 8, 8, 7, 5, 0, 4],
          [10,10,10,10, 9, 9, 5, 5, 3, 3, 8, 8, 7, 5, 4, 0]]

''' Vector b1 such that D_17_1 = [[A16, b1], [b1^T, 0]] '''
b1 = [8, 8, 8, 8, 7, 7, 5, 5, 5, 5, 6, 6, 5, 3, 4, 4]

''' Vector b2 such that D_17_2 = [[A16, b2], [b2^T, 0]] '''
b2 = [5, 5, 5, 5, 6, 6, 8, 8, 8, 8, 3, 5, 4, 4, 7, 7]

''' Function that returns the eigenvalues and eigenvectors of a matrix D '''
def eigenval_eigenvec(D):
    return np.linalg.eig(D)[0], np.linalg.eig(D)[1]

''' Eigenvalues and eigenvectors of D_17_1, D_17_2 and D_16 '''
eigenvalues_1, eigenvectors_1 = eigenval_eigenvec(D_17_1)
eigenvalues_2, eigenvectors_2 = eigenval_eigenvec(D_17_2)
eig = eigenval_eigenvec(D_16)

''' Here, we loop over all eigenvectors of A16 and check whether those are orthogonal
to b1 and b2 '''
for i in range(16):
    print('Eigenvalue is ', eig[0][i])
    print('Eigenvector is ', eig[1][:,i])
    print(round(np.inner(eig[1][:,i], b1),5))
    print(round(np.inner(eig[1][:,i], b2),5))
```

# B

# Python code for pairwise distance matrix generation

```
import numpy as np

def pbt(n):
    ''' Here, we use recursion to create the pairwise distance matrix of a perfectly
    balanced tree on n leaves. We split each group of leaves into halves. Every pair
    of leaves has distance 2 * (depth - level + 1), which corresponds to the distance
    to their lowest common ancestor. The pairwise distance matrix is returned, together
    with a list of the depths of all leaves. '''
    depth = np.log2(n)
    D = np.zeros((n, n))
    def fill(D, half, level):
        if len(half) <= 1:
            return
        mid = len(half) // 2
        left, right = half[:mid], half[mid:]

        for i in left:
            for j in right:
                D[i][j] = D[j][i] = 2 * (depth - level + 1)

        fill(D, left, level + 1)
        fill(D, right, level + 1)

    fill(D, list(range(n)), 1)
    depth_list = [depth for i in range(n)]
    return D, depth_list

def cat(n):
    ''' Here, we create the pairwise distance matrix corresponding to a caterpillar
    matrix. All leaves in a caterpillar are connected to a central path. This function
    generates the pairwise distance matrix if the labels are ordered according to the path.
    That means, each next leaf is one more edge away. The pairwise distance matrix is returned,
    together with a list of the depths of all leaves. '''
    D = np.zeros((n,n))
    for i in range(n):
        for j in range(i+1, n):
            if (i == 0 and j == 1) or (i == 1 and j == 0):
                D[i][j] = D[j][i] = 2
```

```
            elif abs(i-j) == 1:
                D[i][j] = D[j][i] = 3
            elif abs(i-j) != 0:
                D[i][j] = D[j][i] = D[i][j-1] + 1
    depth_list = [i for i in range(n - 1, 0, -1)]
    depth_list.insert(0, n - 1)
    return D, depth_list

def pairwise_matrix_generator(D1, d1, D2, d2):
    ''' This function merges two phylogenetic trees, with corresponding pairwise
    distance matrices D1 and D2, and d1 and d2 are lists with the depths of each leaf.
    We embed D1 in the top-left block and D2 in the bottom-right block. Then the
    distance between a leaf in the left tree and a leaf in the right tree is the sum
    of their depths, plus 2 (the edges that join the roots of the subtrees).
    The pairwise distance matrix is returned, together with a list of the depths of all
    leaves.'''
    n1, n2 = D1.shape[0], D2.shape[0]
    D = np.zeros((n1+n2, n1+n2))
    D[:n1,:n1], D[n1:,n1:] = D1, D2
    for i in range(n1):
        for j in range(n2):
            D[i][n1 + j] = D[n1 + j][i] = d1[i] + d2[j] + 2
    depth_list = d1 + d2
    depth_list = [x+1 for x in depth_list]
    return D, depth_list


def generate_matrices(n):
    ''' In this function, we start with a single leaf (cat(1)) and use recursion to
    generate all possible pairwise distance matrices on n leaves.
    We first used the other initializations (n=2,3,4) as well, but they were not necessary.
    All possible matrices are returned, together with their depth lists.

    Note that the output might contain duplicates, as trees with the same sub-tree on both
    sides are counted twice. '''

    if n == 1:
        D, d = cat(1)
        return [D], [d]
#     if n == 2:
#         D, d = cat(2)
#         return [D], [d]
#     if n == 3:
#         D, d = cat(3)
#         return [D], [d]
#     if n == 4:
#         D1, d1 = cat(4)
#         D2, d2 = pbt(4)
#         return list(zip(*[(D1, d1), (D2, d2)]))

    all_matrices = []

    for i in range(1,n//2+1):
        left_matrices, left_depths = generate_matrices(i)
        right_matrices, right_depths = generate_matrices(n-i)
```

```
        for left_matrix, left_depth in zip(left_matrices, left_depths):
            for right_matrix, right_depth in zip(right_matrices, right_depths):
                new_matrix, new_depth = pairwise_matrix_generator(left_matrix, left_depth,
                                                                  right_matrix, right_depth)
                all_matrices.append((new_matrix, new_depth))

    matrices, depths = zip(*all_matrices)
    return matrices, depths


n = int(input("Enter the number of leaves: "))  # Enter the desired number of leaves
for i in generate_matrices(n)[0]:               # Loops over all pairwise distance matrices
    print('Pairwise distance matrix is \n', i)         # Prints the pairwise distance matrix
    print('Corresponding spectrum: ', np.linalg.eigh(i)[0])  # Prints the corresponding spectrum
    print('')



''' Before creating the generate_matrices function, we looked at trees on n <= 6 leaves
and figured out how they could be created using smaller trees (either perfectly balanced trees
or caterpillars. '''

# All possibilities for 4 leaves
cat4, dcat4 = cat(4)
pbt4, dpbt4 = pbt(4)

# All possibilities for 5 leaves
cat5, dcat5 = cat(5)
D5_1, d5_1 = pairwise_matrix_generator(pbt(2)[0], pbt(2)[1], cat(3)[0], cat(3)[1])
D5_2, d5_2 = pairwise_matrix_generator(pbt4, dpbt4, cat(1)[0], cat(1)[1])

# All possibilities for 6 leaves
cat6, dcat6 = cat(6)
D6_1, d6_1 = pairwise_matrix_generator(D5_1, d5_1, cat(1)[0], cat(1)[1])
D6_2, d6_2 = pairwise_matrix_generator(D5_2, d5_2, cat(1)[0], cat(1)[1])
D6_3, d6_3 = pairwise_matrix_generator(cat(3)[0], cat(3)[1], cat(3)[0], cat(3)[1])
D6_4, d6_4 = pairwise_matrix_generator(pbt(2)[0], pbt(2)[1], cat(4)[0], cat(4)[1])
D6_5, d6_5 = pairwise_matrix_generator(pbt(2)[0], pbt(2)[1], pbt(4)[0], pbt(4)[1])
```

# C

# Characteristic polynomials of $n \times n$ matrices

Here, the characteristic polynomial for a $3 \times 3$ matrix is given.

Let $D(T) = \begin{bmatrix} 0 & d(l_1,l_2) & d(l_1,l_3) \\ d(l_2,l_1) & 0 & d(l_2,l_3) \\ d(l_3,l_1) & d(l_3,l_2) & 0 \end{bmatrix}$. Then

$$\det(D(T)-\lambda I) = \begin{vmatrix} -\lambda & d(l_1,l_2) & d(l_1,l_3) \\ d(l_2,l_1) & -\lambda & d(l_2,l_3) \\ d(l_3,l_1) & d(l_3,l_2) & -\lambda \end{vmatrix} = -\lambda \begin{vmatrix} -\lambda & d(l_2,l_3) \\ d(l_3,l_2) & -\lambda \end{vmatrix} - d(l_1,l_2) \begin{vmatrix} d(l_2,l_1) & d(l_2,l_3) \\ d(l_3,l_1) & -\lambda \end{vmatrix}$$
$$+ d(l_1,l_3) \begin{vmatrix} d(l_2,l_1) & -\lambda \\ d(l_3,l_1) & d(l_3,l_2) \end{vmatrix}.$$

Each $2 \times 2$ determinant can be computed using the formula given in Section 2.4.

Similarly, for a general $n \times n$ matrix, let $D(T)$ be given by

$$D(T) = \begin{bmatrix} 0 & d(l_1,l_2) & \dots & d(l_1,l_n) \\ d(l_2,l_1) & 0 & \dots & d(l_2,l_n) \\ \vdots & \vdots & \ddots & \vdots \\ d(l_n,l_1) & d(l_n,l_2) & \dots & 0 \end{bmatrix}.$$

Then

$$\det(D(T)-\lambda I) = \begin{vmatrix} -\lambda & d(l_1,l_2) & \dots & d(l_1,l_n) \\ d(l_2,l_1) & -\lambda & \dots & d(l_2,l_n) \\ \vdots & \vdots & \ddots & \vdots \\ d(l_n,l_1) & d(l_n,l_2) & \dots & -\lambda \end{vmatrix} = -\lambda \begin{vmatrix} -\lambda & d(l_2,l_3) & \dots & d(l_2,l_n) \\ d(l_3,l_2) & -\lambda & \dots & d(l_3,l_n) \\ \vdots & \vdots & \ddots & \vdots \\ d(l_n,l_2) & d(l_n,l_3) & \dots & -\lambda \end{vmatrix}$$
$$-d(l_1,l_2) \begin{vmatrix} d(l_2,l_1) & d(l_2,l_3) & \dots & d(l_2,l_n) \\ d(l_3,l_1) & -\lambda & \dots & d(l_3,l_n) \\ \vdots & \vdots & \ddots & \vdots \\ d(l_n,l_1) & d(l_n,l_3) & \dots & -\lambda \end{vmatrix} + \dots + (-1)^{n+1} d(l_1,l_n) \begin{vmatrix} d(l_2,l_1) & -\lambda & \dots & d(l_2,l_{n-1}) \\ d(l_3,l_1) & d(l_3,l_2) & \dots & d(l_3,l_{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ d(l_n,l_1) & d(l_n,l_2) & \dots & d(l_n,l_{n-1}) \end{vmatrix},$$

Note that each determinant in the formula can again be computed using the same formula.