Title
Graceful Tree Labelings

Author
Xavier Martí Bofill

Advisor
Mª del Camino Teófila Balbuena Martínez

Department
Matemàtica Aplicada III

Date
October 2012
# Index

1. **Introduction**  
   5

2. **Graphs, trees and graceful labelings**  
   8
   - 2.1. Graphs  
   - 2.2. Labelings  
   - 2.3. Trees  
   - 2.4. Background. The Ringel-Kotzig’s conjecture  
   8

3. **Advances**  
   16
   - 3.1. Labeling algorithms  
   - 3.2. Formation algorithms  
   - 3.3. Computational algorithms  
   16

4. **Adjacency matrices**  
   39
   - 4.1. Description and representation  
   - 4.2. General properties  
   - 4.3. Adjacency matrices for trees  
   - 4.4. Labeling changes  
   39

5. **Graceful adjacency matrices**  
   43
   - 5.1. Matricial representation for graceful labelings  
   - 5.2. Adjacency matrices for proven graceful tree classes  
   - 5.3. Coding  
   43

6. **Matricial unions for graceful trees**  
   52
   - 6.1. Bipartite labelings and union requirements  
   - 6.2. Edge/Vertex identification  
   - 6.3. Connection  
   - 6.4. Multiple matrix union  
   53

7. **Trees having a perfect matching**  
   68
   - 7.1. Introduction  
   - 7.2. Strong gracefulness  
   - 7.3. Reduction of the problem  
   - 7.4. Equivalent conjecture  
   68

8. **Conclusions**  
   77

9. **References**  
   79
1. Introduction

The Graceful Tree Conjecture was published almost 40 years ago, and nowadays it is still an open problem. It seems as difficult to solve as easy it is to enunciate: *every tree admits a vertex labeling on 0, ..., n – 1 such that the set of absolute values of the differences of the numbers assigned to the ends of each edge is the set {1, 2, ..., n – 1}*. Kotzig and Rosa have referred to the graceful tree conjecture as a "graphical disease" due to the inability to establish results reaching further than showing that small classes of trees are graceful.

Professor Camina Balbuena of my University (UPC, Barcelona) offered me the possibility of having her tutoring my undergraduate thesis on this problem. I found it to be interesting and challenging, and since then I have done a vast research on the matter.

This Thesis was meant to be a review of research done in the field of graceful labelings for trees. And for Chapters 2 and 3, it is. Basic concepts about graph labeling are explained in Chapter 2, and also how the graceful tree conjecture was established. Chapter 3 analyses some of the efforts made until today to solve the Graceful Tree Conjecture.

But somewhere along the way, I decided to look at the problem from a matricial angle, surprised that almost no one had already used adjacency matrices to study graceful labelings for trees. I found out there were some pros and cons in using them, given that while some properties of the trees are best seen graphically, adjacency matrices allow a greater abstraction.

Chapters 4 and 5 explain the basics for using adjacency matrices, including some properties about adjacency matrices representing graceful labelings that I defined, only to find out it had already been published, although not as a paper but as a bachelor of science.

Chapter 6 is my real contribution to the problem. I establish a procedure to join trees that enables the creation and graceful labeling of huge trees, regardless of its properties. Given that graceful labeling complexity increases factorially with the tree size, when size of a tree surpasses 30 vertices it becomes a real problem to gracefully label it, even with a computer. Further research on this method should be done to control properly the way trees are joined.
Finally, Chapter 7 applies adjacency matrices to trees having a perfect matching, proving that the conjecture “every tree having a perfect matching is strongly graceful” is equivalent to the Graceful Tree Conjecture. This had already been proven, but the paper is not easily accessible and probably it was not proven the same way.

My first conclusion would be that I do not see this problem close to be solved. However, work continues on the conjecture, more or less in the same direction. For a more comprehensive treatment on graceful labelings, the reader is encouraged to reference “A Dynamic Survey of Graph Labeling” maintained by Joseph A. Gallian [Ga02].

My greatest gratitude is for my tutor Camino Balbuena who has introduced me to the world of discrete mathematics with this thrilling problem and has offered me all her help in the understanding of it.
2. Graphs, trees and graceful labelings

2.1. Graphs

A graph is a representation of a set of objects and the relations between them. They are used as a scheme for more complex systems in any area of study. However, here we will discuss graphs as a purely abstract concept, without concerning about its possible uses.

Graphs are defined as a set of nodes or vertices $V$ connected between them by a set of edges $E$ so that $E = \{(i, j), i, j \in V\}$. Let $n$ be the cardinality of $V$ and let the graph be named as $G(V, E)$. Figure 1 depicts the drawing of a graph on 14 vertices and 15 edges.

![Graph on 14 vertices and 15 edges](image)

Every vertex $u$ will share edges with a subset of vertices in the graph, and these vertices will be called its neighbours. The amount of neighbours a vertex $u$ has is called its degree $\delta(u)$.

If the edge $(i, j)$ belongs to $E$ for every $i, j \in V$ then the graph is called a complete graph and it is noted by $K_n$, $n$ being the amount of vertices. If there is a subset of vertices connected by edges such that one can travel through these edges stepping on every vertex only once and reaching in the end the initial vertex, this is called a cycle. If there is no path of edges in $G$ that connect any $i, j \in V$, $G$ will be a disconnected graph.
Unidirectional graphs can be defined as those whose edges work like \( i \rightarrow j \), but here we will only study bidirectional graphs, whose edges work both ways, meaning that \((i,j) \in E\) implies \(i \rightarrow j\) and \(j \rightarrow i\).

Another important property about graphs is bipartity. A graph is bipartite if vertices can be classified in 2 subsets in such a way that there is no edge in \(E\) with both its vertices from the same subset. This means that all neighbours are in a different subset than the own. This will be a remarkable property in the following chapters.

Calculus complexity in graph related studies escalate in a factorial proportion with the number of vertices in the graph. This leads to the necessity to develop different techniques to avoid raw calculus, as it would imply unbearable computational costs. This is the main reason that graphs are the subject of a vast range and amount of studies, regarding its behaviours, its properties and ways to process them.

2.2. Labelings

Graph labelings are defined as a bijective function \( f(V) = N \) or \( f(E) = N \), being \( N \) a set of natural numbers, each corresponding to one different vertex or edge. We will only consider labelings whose labels go from 0 to \( n - 1 \) and do not miss any number in between, where \( n \) is the number of vertices. This way, there will not be any two vertices nor two edges with the same label. Figure 2 shows a graph unlabeled, with an edge labeling and with a vertex labeling.

![Figure 2. Example of graph labeling](image)

A labeling \( f(V) = N \) is graceful if the absolute value \(|f(u) - f(v)|\) is assigned to the edge \((u, v)\) as its label and the resulting edge labels are mutually distinct.
An example is shown in Figure 3.

![Figure 3. A graph with a grafcul labeling](image)

Not every graph admits a graceful labeling. In fact, any graph with the same number of edges than vertices or more will not admit a graceful labeling, unless we allow vertex labels higher than the number of vertices in the graph.

Another used labeling is the harmonic labeling, basically the same as graceful but edge labels are $|f(u) + f(v)|$, where the sum is taken modulo $n$. Finally there is also graph colouring, consisting in assigning a limited number of labels to all vertices in such a way that no edge can connect two vertices with the same labels. Notice that bipartite graphs and 2-label colourable graphs are the same.

### 2.3. Trees

We define a tree as a graph with no cycles within. this means that from any vertex $u$ there is only one path to go to any other vertex $v$, without stepping on any vertex more than once. This implies that the total amount of edges in a tree is $m = n - 1$. Figure 4 shows an example of a tree.
Examples of trees can be found in genealogical trees, organization hierarchies, distributional models or data nets configurations.

To describe trees we will need some concepts explained below:

- **Leafs** are vertices with only one neighbor, $\delta(u) = 1$. Every tree has at least 2 leafs.
- A **path** is a tree every vertex of which has degree two, excepting two leafs, the beginning and the end of the path.
- **Stars** are those trees where every vertex is a leaf except the central one $u$, with $\delta(u) = n - 1$.
- We define $d(i,j)$ as the **distance** between vertices $i$ and $j$, counting the number of edges forming the path within the tree that connects them.
- If we sever an edge of our tree we get two **subtrees**, and every subtree has the same properties than a tree.
- The **root** of a tree is a vertex chosen in an arbitrary way concerning concepts of symmetry with the rest of the tree. Usually if we count the maximum distance from a vertex to any of the other vertices of the tree, the root will give the lowest value. A tree with a root is called a **rooted tree**. A root can be simple or double.
- Every subtree acquired severing an edge of the root is called a **branch**.
- Every **level** $L_x$ of a rooted tree is formed by every vertex at distance $x$ from the root of the tree.
- **Every tree is bipartite**. This property is shown classifying every vertex alternatively as we move by edges, because the inexistence of cycles makes impossible the concurrence of two vertex of the same class in an edge. In a more mathematical expression, we can classify concerning distance from an arbitrary vertex $j$, becoming class $A$ every vertex $i$ such that $d(i,j) = odd$, and class $B$ every vertex $i$ such that $d(i,j) = even$. 
2.4. Background. Ringel-Kotzig’s conjecture

In 1963, Ringel formulated the following problem, which has been ever since known as the Ringel’s conjecture [Ros67]:

“Every tree containing \( m + 1 \) vertices decomposes \( K_{2m+1} \).”

\( K_{2m+1} \) is a complete graph containing \( 2m + 1 \) vertices. This means that \( K_{2m+1} \) can be decomposed in any \( 2m + 1 \) isomorphical trees, every edge of the complete graph represented in only one of such trees.

This conjecture remains unsolved nowadays. Rosa [Ros67], exposed that Kotzig made a stronger conjecture, Kotzig conjecture says:

“Every tree containing \( m + 1 \) vertices cyclically decomposes \( K_{2m+1} \).”

This means that if we position every vertex of the complete graph identifying the vertices of a regular polygon, every tree containing \( m + 1 \) vertices can be positioned in such a way that if we rotate one vertex of the tree to every other vertex in the complete graph keeping the relative position of the other vertices in the tree we will obtain every edge in the complete graph repeating none.

In 1967, Rosa published the document [Ros67] intending an approximation to the Ringel-Kotzig’s conjecture’s resolution. His idea was to use a labeling of the vertices of a graph \( H \) of order \( m \) in order to prove it could cyclically decompose \( K_{2m+1} \). He made reference to a labeling system as a graph’s valuation. We define \( O_G \) as a labeling of \( G \). We will denote \( V_{OG} \) as the set of labels assigned to the vertices of \( G \) and \( E_{OG} \) as the set of labels assigned to the edges of \( G \) induced by \( O_G \), becoming these, as usual, the absolute difference between the labels of the two vertices connected by the edge.

Consider the following conditions:

(1) \( V_{OG} \subseteq \{1, 2, \ldots, n\} \),

(2) \( V_{OG} \subseteq \{1, 2, \ldots, 2n\} \),

(3) \( E_{OG} = \{1, 2, \ldots, n - 1\} \),

(4) \( E_{OG} = \{x_1, x_2, \ldots, x_n\} \), where \( x_i = i \) or \( x_i = 2n + 1 - i \),

(5) There is \( x \in \{1, 2, \ldots, n\} \), so that for any edge \( v_i v_j \) is provided that \( a_i \leq x < a_j \) or \( a_j \leq x < a_i \), where \( a_k \) is the label assigned to \( v_k \) for any \( k \).
Given these conditions Rosa defines four kinds of labelings:

A $\rho$ – valuation satisfies conditions (2) and (4).

A $\sigma$ – valuation satisfies conditions (2) and (3).

A $\beta$ – valuation satisfies conditions (1) and (3).

An $\alpha$ – valuation satisfies conditions (1), (3) and (5).

Rosa’s $\beta$ – valuation is what we now call a graceful labeling (the term was first introduced by Golomb in [Gol72]. An $\alpha$ – valuation is also known as a bipartite labeling, given that if all vertices are classified in two classes in such a way that there is no edge connecting two vertices of the same class, the resulting classes will include the sets of labels $1, 2, ..., i$ and $i + 1, i + 2, ..., n$, respectively.

The former definitions also suggest a labeling hierarchy: ordered from stronger to weaker restrictions, we have $\alpha, \beta, \sigma$ and $\rho$ – valuation, and every labeling is a special case of its successor in the hierarchy. This means that if we could prove that every tree admits a bipartite labeling, the graceful tree conjecture would follow. However, Rosa already proved that not every tree admits a bipartite labeling and was capable of identifying a family of trees that does not admit it.

![Figure 5. Tree without possible bipartite labeling](image)

An example of a tree without possible bipartite labeling is the tree $T$ shown in Figure 6. It was proven by Bloom [Blo79] in 1979 that it does not admit this kind of labeling. It is also the smallest tree that is not a caterpillar, so it follows that it is the smallest tree not to admit a bipartite labeling, given that every caterpillar admits a bipartite labeling as we will see later.
In his paper [Ros67], Rosa also published the following statement, which ended up being the beginning of the graceful tree conjecture:

Theorem: If $H$ is a graceful graph containing $m$ edges, then $K_{2m+1}$ is $H$–decomposable. In fact, $K_{2m+1}$ can be cyclically decomposed into copies of $H$.

Proof. (As it appears in [EH06])

As $H$ is graceful, there is a graceful labeling of $H$, that is to say, vertices in $H$ can be labelled by a subset $\{0, 1, \ldots, m\}$ in such a way that induced edge labels become $1, 2, \ldots, m$. Let

$$V(K_{2m+1}) = \{v_1, v_2, \ldots, v_{2m+1}\},$$

where vertices in $K_{2m+1}$ are organized in a regular polygon of $2m + 1$ vertices, becoming the cycle $C_{2m+1}$. Every vertex labelled $i$ ($0 \leq i \leq m$) in $H$ is positioned in $v_i$ in $K_{2m+1}$. Every edge of $H$ is drawn as a straight segment in $K_{2m+1}$, which denotes the resulting copy of $H$ as $H_1$. Therefore, $V(H_1) \subseteq \{v_1, v_2, \ldots, v_{m+1}\}$.

Each edge $v_s v_t$ of $K_{2m+1}$ ($1 \leq s, t \leq 2m+1$) is labeled as $d_c(v_s, v_t)$, the number of edges in the shorter path in $C_{2m+1}$ between $v_s$ and $v_t$, so $1 \leq d_c(v_s, v_t) \leq m$. Therefore, $K_{2m+1}$ contains exactly $2m + 1$ edges with label $i$ for every $i \in \{1, 2, \ldots, m\}$ and $H_1$ contains exactly one edge labeled $i$ for every $i \in \{1, 2, \ldots, m\}$. Note that when an edge of $H_1$ is rotated a certain angle (for example, clockwise) of $2\pi k/(2m + 1)$ radians, where $1 \leq k \leq m$, the result is an edge with the same label. We denote the subgraph obtained by rotating $H_1$ a certain angle clockwise $2\pi k/(2m + 1)$ radians as $H_{k+1}$ for every $1 \leq k \leq 2m + 1$. Then, $H_{k+1}$ is isomorphic to $H$, and $H_k$ and $H_1$ have a different subset of edges if $k \neq l$. Therefore, $H_1, H_2, \ldots, H_{2m+1}$ provide a cyclical decomposition of $K_{2m+1}$ into $2m + 1$ copies of $H$.

In fact, a graph $H$ of $m$ edges does not need to be graceful in order to cyclically decompose $K_{2m+1}$. In his paper, Rosa also specifies exactly when $H$ cyclically decomposes $K_{2m+1}$.

Theorem. A subgraph $H$ of $m$ edges cyclically decomposes $K_{2m+1}$ if and only if a $\rho$–valuation of $H$ exists.

Rosa’s research changed the goal of every effort on demonstrating Ringel’s conjecture into demonstrating that every tree is graceful.
3. Advances

Since Ringel, Kotzig and Rosa formulated the problem, many papers have been published giving further insight on the subject. Most of them have been led to prove that every tree is graceful under certain restrictions, that is to say, that certain classes of trees are graceful. In this document are represented the most significant advances.

3.1. Labeling algorithms

We define as labeling algorithms those algorithms which given a tree under certain restrictions a labeling method is provided to gracefully label any tree of this class.

Caterpillars

Caterpillars are defined as those trees such that if one removes all of its leaves, the remaining graph is a path. In his paper [Ros67], Rosa proved all caterpillars to be graceful, the first class to be proven graceful and possibly the most useful one when it comes to prove other classes to be graceful. To accomplish the proof, he used a basic lemma of the graceful trees:

Lemma: Let $T$ be a graceful tree of $n$ vertices with a graceful labeling function $f: V(T) \rightarrow \{0, 1, \ldots, m\}$, $m = n - 1$. Then, the labeling function $f': V(T) \rightarrow \{0, 1, \ldots, m\}$ such that $f'(v) = m - f(v)$ is also graceful.

Proof: Observe that every induced edge label will remain the same given that $g'(uv) = |f'(u) - f'(v)| = |m - f(u) - m + f(v)| = |f(v) - f(u)| = g(uv)$.

One can say that by applying this lemma we are performing a labeling inversion.

From here on, the labeling algorithm for caterpillars is simple. First, we must identify the vertices forming the principal path (longest possible path, it may not be unique) $v_1, v_2, ..., v_k$, $k < n$, $v_1$ and $v_k$ being leafs, also known as head and tail of the caterpillar. Let 0 be the label for $v_1$ and let $m$ be the label for $v_2$. After that, we assign the smallest possible labels to the neighbours of $v_2$, leaving the higher of these ones to $v_3$. Once we reach this point we perform a labeling inversion and we repeat the step for the next vertex, and so on until we reach the tail. This process guarantees a graceful and bipartite labeling ($\alpha - valuation$). By way of example see Figure 6.
Spiders

A spider is defined as a set of paths joined by a central vertex the degree of which has to be higher than two (otherwise it would be a path). Such paths are the legs of the spider. Only certain kinds of spiders have been proven as graceful.

One of the most important algorithms to label a family of spiders is the one that labels every spider with \( l \) legs of lengths between \( m \) and \( m+1 \). It is shown in [BLW10] that this kind of spiders are graceful. Here it is reproduced their proof.

We may assume that \( l \geq 3 \), as otherwise \( T \) consists of a path, which is known to be graceful.

**Case 1.** (\( l \) is odd) Let \( l = l_0 + l_1 \), where \( l_i \) is the number of legs of length \( m+i \) for \( i \in \{0, 1\} \).

Note that \( T \) has \( n+1 = l \cdot m + l_1 + 1 \) vertices, to be labeled by the set \( \{1, 2, \ldots, n\} \). Label the legs by \( L_1, L_2, \ldots, L_l \) so that \( L_{l_1}, L_{l_2}, \ldots, L_{l_1} \) have length \( m+1 \) and \( L_{l_1+1}, \ldots, L_{l_i} \) have length \( m \).

Let \( v^* \) denote the branch point of \( T \) and denote by \( v_{i,j} \) the vertex in \( L_i \) of distance \( j \) from \( v^* \).

Let \( \phi \) be the labeling defined as follows:

(i) \( \phi(v^*) = 0 \),

(ii) if \( i \) and \( j \) are both odd, \( \phi(v_{i,j}) = n - \frac{\frac{i-1}{2} - \frac{(i-1)l}{2}}{2} \),

(iii) if \( i \) and \( j \) are both even, \( \phi(v_{i,j}) = n - \frac{\frac{i-1}{2} - \frac{i}{2} - \frac{(j-2)l}{2}}{2} \),

(iv) if \( i \) is even and \( j \) is odd, \( \phi(v_{i,j}) = \frac{i}{2} + \frac{\frac{(j-1)l}{2}}{2} \), and

(v) if \( i \) is odd and \( j \) is even, \( \phi(v_{i,j}) = \frac{i-1}{2} + \frac{i+1}{2} + \frac{(j-2)l}{2} \).
The labeling \( \phi \) places 0 at the spider’s center and, traversing the longer legs first, alternates between the highest and the lowest remaining unused labels, spiralling away from the center.

To help compute the induced edge labels, note that \( \phi \)'s local maxima occur at \( v_{i,j} \) for which \( i \equiv j \pmod{2} \), and for such \( i \) and \( j \),

\[
\phi(v_{i,j}) - \phi(v_{i,j+1}) = n - \frac{l - 1}{2} - i + (1 - j)l > 0
\]

and

\[
\phi(v_{i,j}) - \phi(v_{i,j-1}) = n - \frac{l - 1}{2} - i + (2 - j)l > 0
\]

Suppose, in order to obtain a contradiction, that \((i, j) \neq (i', j')\), \( i \equiv j \pmod{2} \), \( i' \equiv j' \pmod{2} \), and \( \phi(v_{i,j}) - \phi(v_{i,j+1}) = \phi(v_{i',j'}) - \phi(v_{i',j'+1}) \). From the equations above we obtain

\[
i - i' + (j - j')l = 0 \Rightarrow l = \frac{i - i'}{j - j'}
\]

Note that \( j \neq j' \), since otherwise \( i = i' \) as well, and \((i, j) = (i', j')\) after all. Thus \( |i - i'| < l \) and \( |j - j'| \geq 1 \) and

\[
l = \frac{i - i'}{j - j'} < \frac{l}{1} = l
\]

becomes a contradiction.

Similar contradictions arise should \( \phi(v_{i,j}) - \phi(v_{i,j+1}) = \phi(v_{i',j'}) - \phi(v_{i',j'+1}) \) or \( \phi(v_{i,j}) - \phi(v_{i,j-1}) = \phi(v_{i',j'}) - \phi(v_{i',j'-1}) \) hold. Thus no edges bear the same labels, and \( \phi \) is graceful.

An example of this procedure is provided in Figure 7.

**Case 2.** \((l \text{ is even})\) Without loss of generality assume \( L_i \) is a leg of length \( m \). Remove it, resulting in a tree \( T_0 \) with an odd number, \( l - 1 \), of legs. The construction above yields a graceful labeling \( \phi_0 \) of \( T_0 \) such that \( \phi_0(v*) = 0 \). Let \( |V(T_0)| = n' + 1 \). A new graceful labeling is defined, \( \phi' \), on \( T_0 \) by \( \phi'_0(v) = n' - \phi_0(v) \) for all \( v \).

Construct a new tree \( T_1 \) by appending a new vertex, \( w_1 \), to \( T_0 \)'s center. Define \( \phi_1 \) on \( V(T_1) \) by \( \phi_1(w1) = 0 \) and \( \phi_1(v) = \phi'_0(v) + 1 \) for all \( v \in V(T_0) \). Define \( \phi'_1 \) on \( T_1 \) by \( \phi'_1(v) = n' + 1 - \phi_1(v) \) for all \( v \); note \( \phi'_1(w_1) = n' + 1 \).
We may now append a vertex \( w_2 \) to \( w_1 \) and construct graceful labelings \( \phi_2 \) from \( \phi'_1 \) and \( \phi'_2 \) from \( \phi_2 \), and so forth, until we have reconstructed \( L_t = \{w_1, w_2, ..., w_m\} \), recovering \( T \).

**Remark.** The proof of Case 2 actually shows gracefulness for any tree formed by appending an extra leg of any length to an odd-legged spider with legs of two lengths differing by at most 1.

![Figure 7. Odd legged spider gracefully labeled](image)

### 3.2. Formation algorithms

Another kind of algorithms to prove to be graceful certain classes of trees are the formation algorithms. These ones use to start with a known graceful tree, already with a graceful labeling, and modify it, always keeping its gracefulness, until reaching the required tree. This process is normally done by transferring edges from one vertex to another, as we are going to see now.
Symmetrical trees

Definition. A symmetrical tree is a rooted tree in which every level contains vertices of the same degree. Figure 8 shows a symmetrical tree in which every vertex different from a leaf has degree three.

It has been shown in [BS76] as well as in [Rob11] that all symmetrical trees are graceful. Here Proof by [Rob11] is represented.

Theorem. All symmetrical trees are graceful.

Proof. The author shows by induction on the number of layers that all symmetrical trees are graceful and there exists a graceful labeling which assigns the number 1 to the root.

If $T$ is a symmetrical tree with $0$ layers, then, it consists of 0 edges and just one vertex, and clearly there is a graceful labeling which assigns 1 to that vertex. Suppose we have proved that for some $l > 0$ all symmetrical trees with $\leq l - 1$ layers are graceful and each of them has a graceful labeling which assigns the number 1 to the root.

Idea of Induction Step. The idea of the induction step is to consider a rooted symmetrical tree for which we know that its $k$ children $T_1, T_2, ..., T_k$ are graceful (and isomorphic to each other). The children are labeled with their (identical) graceful labelings and then add certain numbers to each of the vertices. The way this is done the following. The children are ordered from left to right. Then, if $n$ is the number of vertices in each child, start from the 0th layer of the children and add $(k - 1)n$ to the root of $T_1$, $(k - 2)n$ to the root of $T_2$, ..., and 0 to the root of
the \( k \)-th one. Then, for the first layer, start from right to left this time and add \((k - 1)n\) to each of the vertices in the 1st layer of \( T_k \), then, we add \((k - 2)n\) to each of the vertices in the 1st layer of \( T_{k-1} \), \ldots, and 0 to each of the vertices in the first layer of \( T_1 \). So, then go on with the second layer and we start from left to right, and so on until finishing with the last layer. Then, write \( nk + 1 \) on the root of the new tree. Then, apply the transformation \( x \rightarrow nk + 2 - x \) to each of the vertices, so that there is a 1 at the root and the resulting labeling, as shown in the sequel, is graceful. Figure 9 shows all the steps of this procedure.

![Figure 9. All the steps in a graceful labeling of a symmetrical tree](image)

Let \( T \) be a symmetrical tree with \( l \) layers. Let \( v \) be the root, \( v_1, v_2, \ldots, v_k \) its children, and \( T_1, T_2, \ldots, T_k \) the corresponding rooted subtrees. Then, by the previous lemma, it is known that \( T_1, T_2, \ldots, T_k \) are symmetrical trees and are isomorphic as rooted trees. Let \( \varphi_i: T_1 \rightarrow T_i \) be the isomorphisms as constructed in the lemma. So, in particular, \( \varphi_i \) sends the \( l \)-th level of \( T_1 \) to the \( l \)-th level of \( T_i \) and for each vertex \( u \in T_1, \varphi_i \) sends the children of \( u \) to the children of \( \varphi_i(u) \). By the induction hypothesis, there exists a graceful labeling \( f_1: T_1 \rightarrow \{1, 2, \ldots, n\} \), where \( n = |V(T_1)| \) such that \( f_1(v_1) = 1 \). So, \( f_1 \) induces graceful labelings \( f_i = f_1 \circ \varphi_i^{-1}: T_i \rightarrow \{1, 2, \ldots, n\} \) such that \( f_i(v_i) = 1 \) for each \( i = 2, 3, \ldots, k \). For convenience of notation define \( \varphi_1 = ld_{T_1}, \) so, \( f_1 = f_1 \circ \varphi_1 \).
Now, consider the following labeling $f: T \rightarrow \{1, 2, \ldots, nk + 1\}$. (Note that $|V(T)| = nk + 1$).

Let $f(v) = nk + 1$ and $f(v_i) = 1 + (i - 1)n$. Let the vertices on level $l$ of $T_i$ be $u_{1,1}, u_{1,2}, \ldots, u_{1,m_i}$. So, the vertices on level $l$ of $T_i$ are $\varphi_i(u_{1,1}), \varphi_i(u_{1,2}), \ldots, \varphi_i(u_{1,m_i})$. If $l > 1$ is odd, let $f(\varphi_i(u_{1,j})) = f_i(\varphi_i(u_{1,j})) + (i - 1)n = f_i(u_{1,j}) + (i - 1)n$ and if $l > 2$ is even, let $f(\varphi_i(u_{1,j})) = f_i(\varphi_i(u_{1,j})) + (k - i)n = f_i(u_{i,j}) + (k - i)$.

Then, first of all, the edges from the root to its children are assigned the labels: $|f(v) - f(v_i)| = |nk + 1 - (i - 1)n| = n(k - i + 1)$ for $i = 1, 2, \ldots, k$, so, these are exactly the numbers $n, 2n, \ldots, kn$. Consider two adjacent vertices $u, u' \in T_i$ and the corresponding $\varphi_i(u), \varphi_i(u') \in T_i$. Let $u$ be in level $l$ and $u'$ in level $l + 1$.

Then, if $l$ is odd, we have

$$\{g(\varphi_i(u)\varphi_i(u')): i = 1, 2, \ldots, k\} = \{|f(\varphi_i(u)) - f(\varphi_i(u')): i = 1, 2, \ldots, k\}$$

$$= \{|f_i(u) + (k - i)n - f_i(u') - (i - 1)n: i = 1, 2, \ldots, k\}$$

$$= \{|f_i(u) - f_i(u') + (k - 2i + 1)n: i = 1, 2, \ldots, k\}$$

If $l$ is even, then

$$\{g(\varphi_i(u)\varphi_i(u')): i = 1, 2, \ldots, k\} = \{|f(\varphi_i(u)) - f(\varphi_i(u')): i = 1, 2, \ldots, k\}$$

$$= \{|f_i(u) + (i - 1)n - f_i(u') - (k - i)n: i = 1, 2, \ldots, k\}$$

$$= \{|f_i(u) - f_i(u') - (k - 2i + 1)n: i = 1, 2, \ldots, k\}$$

$$= \{|f_i(u) - f_i(u') + (k - 2i + 1)n: i = 1, 2, \ldots, k\}$$

Note that for $1 \leq |m| \leq n - 1$, there is

$$\{m + (k - 2i + 1)n: i = 1, 2, \ldots, k\} =$$

$$\{m + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2}\} \cup$$

$$\{-m - (k - 2i + 1)n: \frac{k + 1}{2} < i \leq k\} \cup \{|m|: 2i =$$

$$= k + 1, i \in \mathbb{N}\} = \{m + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2}\} \cup$$

$$\{-m - (k - 2(k + 1 - i) + 1)n: 1 \leq i < \frac{k + 1}{2}\} \cup \{|m|: 2i = k + 1, i \in \mathbb{N}\} =$$

$$= \{m + (k - 2i + 1)n: 16 \leq 2\} \cup$$

$$\{-m + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2}\} \cup \{|m|: 2i = k + 1, i \in \mathbb{N}\} =$$
= \left\{ |m| + (k - 2i + 1): 16i < \frac{k + 1}{2} \right\} \cup \\
\{n - |m| + (k - 2i): 1 \leq i < \frac{k + 1}{2}\} \cup \{|m|: 2i = k + 1, i \in \mathbb{N}\}.

Now, let 1 \leq m \leq n - 1 be such that m \neq n - m. So, there are two edges uu' \in E(T_1) and ww' \in E(T_1) such that \(|f_1(u) - f_1(u')| = m\) and \(|f_1(w) - f_1(w')| = n - m\). Let u and w be in odd layers of T, so that u' and w' will be in even layers of T. And so, the following two sets are among the labels of T induced by f:

\{ |f_1(u) - f_1(u')| + (k - 2i + 1)n: i = 1, 2, ..., k \} \cup \\
\{ |f_1(w) - f_1(w')| + (k - 2i + 1)n: i = 1, 2, ..., k \} = \\
\{ |f_1(u) - f_1(u')| + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2} \} \cup \\
\{ |f_1(w) - f_1(w')| + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2} \} \cup \\
\{ n - |f_1(u) - f_1(u')| + (k - 2i)n: 1 \leq i < \frac{k + 1}{2} \} \cup \{ |f_1(u) - f_1(u')|: 2i = k + 1 \} \cup \\
\{ |f_1(w) - f_1(w')| + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2} \} \cup \\
\{ n - |f_1(w) - f_1(w')| + (k - 2i)n: 1 \leq i < \frac{k + 1}{2} \} \cup \{ |f_1(w) - f_1(w')|: 2i = k + 1 \} = \\
\{ m + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2} \} \cup \{ n - m + (k - 2i)n: 1 \leq i < \frac{k + 1}{2} \} \cup \{ m: 2i = k + 1 \} \cup \{ n - m: 2i = k + 1 \}.

Now, if k is odd, then k + 1 is even and \{ m + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2} \} \cup \{ m + (k - 2i)n: 1 \leq i < \frac{k + 1}{2} \} \cup \{ m: 2i = k + 1 \} = \{ m + in: 0 \leq i < k \} and if k is even, then, k + 1 is odd, so, \{ m + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2} \} \cup \{ m + (k - 2i)n: 1 \leq i < \frac{k + 1}{2} \} \cup \{ m: 2i = k + 1 \} = \{ m + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2} \} \cup \{ m + (k - 2i)n: 1 \leq i < \frac{k + 1}{2} \} = \{ m + in: 0 \leq i < k \}. Similarly, in both cases \{ n - m + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2} \} \cup \{ n - m + (k - 2i)n: 1 \leq i < \frac{k + 1}{2} \} \cup \{ n - m: 2i = k + 1 \} = \{ n - m + in: 1 \leq i < k \}.

So, for each 1 \leq m \leq n - 1 such that m \neq n - m, the set \{ m + in: 1 \leq i < k \} is among the induced labels for the edges of T.
If \( m = n - m \), then and if \( u, u' \) are such that \( |f_1(u) - f_1(u')| = m \) and \( u \) is in an odd layer and \( u' \) is in an even layer of \( T \), then the following set of labels is among the labels of edges induced by \( f \).

\[
\{|f_1(u) - f_1(u') + (k - 2i + 1)n|: i = 1, 2, ..., k\} = \\
= \{|f_1(u) - f_1(u')| + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2}\} \cup \\
\{n - |f_1(u) - f_1(u')| + (k - 2i)n: 1 \leq i < \frac{k + 1}{2}\} \cup \{|f_1(u) - f_1(u')|: 2i = k + 1\}
\]

\[
= \{m + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2}\} \cup \{n - m + (k - 2i)n: 1 \leq i < \frac{k + 1}{2}\} \cup \\
\cup \{m: 2i = k + 1\} = \\
= \{m + (k - 2i + 1)n: 1 \leq i < \frac{k + 1}{2}\} \cup \{m + (k - 2i)n: 1 \leq i < \frac{k + 1}{2}\} \cup \\
\cup \{m: 2i = k + 1\}.
\]

So, again, if \( k \) is odd, so that \( k + 1 \) is even, this is exactly \( \{m + in: 0 \leq i < k\} \) and if \( k \) is even, so that \( k + 1 \) is odd, again this set is exactly \( \{m + in: 0 \leq i < k\} \).

So, from the edges in \( T_1, T_2, ..., T_k \), we get all labels \( \{m + n: 0 \leq i < k\} \) for \( 1 \leq m \leq n - 1 \) and from the edges between the root \( v \) and \( v_1, v_2, ..., v_k \), all labels \( \{n + in: 0 \leq i \leq k\} \). So, this is a graceful labeling of \( T \). Moreover, \( f(v) = nk + 1 = |V(T)| \), so, if the preliminary lemma is applied and the labels are switched \( f(v) \rightarrow nk + 2 - f(v) \), then, we are going to get \( f(v) = 1 \) and the labeling is still graceful. This completes the induction.

Thus, all symmetrical trees are graceful.

**Trees of diameter 5**

Trees with diameter 2 are star trees and they are instances of caterpillars, hence they are graceful. Rosa proved that trees of diameter at most three are graceful. In 1989 Zhao [Zha89] showed that all trees of diameter 4 are graceful.

In 2001, Hrnčiar and Havija [HH01] showed that all trees of diameter 5 are graceful.

**Lemma.** Let a tree \( T \) with \( n \) edges have a graceful labeling \( f \) and let \( u \in V(T) \) be such that \( f(u) = 0 \) or \( f(u) = n \). Let \( H \) be a caterpillar, \( V(T) \cap V(H) = \emptyset \) and let \( v \in V(H) \) be a vertex which either has a maximal eccentricity or is adjacent to a vertex of maximal eccentricity. If \( T' \)
is the tree obtained by gluing the trees $T$ and $H$ in such a way that the vertices $u$ and $v$ are identified, then, $T'$ is a graceful tree as well.

**Definition.** Let $T$ be a tree and let $uv \in E(T)$. Then, $T_{u,v}$ is the subtree of $T$ induced by the set $V(T_{u,v}) = \{w \in V(T); w = u$ or $v$ is in a $u - w$ path$\}$.

The transformations that Hrnčiar and Haviar use in their paper are based on the following lemma.

**Lemma.** Let $T$ be a tree with a graceful labeling $f$ and let $u$ be a vertex adjacent to the vertices $u_1$ and $u_2$. Let $T'$ be the subtree of $T$ induced by the set $V(T') = (V(T) - (V(T_{u,u_1}) \cup V(T_{u,u_2}))) \cup \{u\}$ and let $v \in V(T')$ with $v \neq u$. Then,

(a) If $u_1 \neq u_2$, $f(u_1) + f(u_2) = f(u) + f(v)$ and the tree $T''$ is obtained by gluing the trees $T_{u,u_1}, T_{u,u_2}$ and $T'$ in such a way that the vertex $v$ of the tree $T'$ is identified with the vertex $u$ of the trees $T_{u,u_1}, T_{u,u_2}$, then $f$ is a graceful labeling. (see Figure 10)

(b) If $u_1 = u_2$, $2f(u_1) = f(u) + f(v)$ and $T''$ is a tree obtained by gluing the trees $T'$ and $T_{u,u_1}$ in such a way that the vertex $v$ of the tree $T'$ is identified with the vertex $u$ of the tree $T_{u,u_1}$, then $f$ is a graceful labeling of the tree $T''$ as well.

\[ \text{Figure 10. Example of transformation (a)} \]

**Proof.** The proof is based on a few simple equalities:

For part (a), $|f(u_1) - f(u)| = |f(u) + f(v) - f(u_2) - f(u)| = |f(v) - f(u_2)|$ and $|f(u_2) - f(u)| = |f(u) + f(v) - f(u_1) - f(u)| = |f(v) - f(u_1)|$, so, the labeling of the new tree is also graceful.
For part (b), \( |f(u_t) - f(u)| = \frac{|f(u) + f(v) - f(u)|}{2} = \frac{|f(u) - f(v)|}{2} \) and \( |f(u_t) - f(v)| = \frac{|f(u) + f(v) - f(v)|}{2} = \frac{|f(v) - f(u)|}{2} \) and so the labeling of the new tree is also graceful.

Using the notation of the Lemma, we are going to say that the tree \( T'' \) is obtained from \( T \) by a transfer of the trees \( T_{u,u_t} \) and \( T_{u,u_2} \) from the vertex \( u \) to the vertex \( v \). For example, from the star on the left of Figure 11, we can get the tree on the right.

![Figure 11. Transfer of edges in a graceful tree](image)

**Definition.** Hrnciar and Haviar [HH01] use two types of transfers of end-edges.

A \( u \rightarrow v \) transfer is a transfer of the first type if the end-vertices of the transferred end-edges have labels \( k, k + 1, \ldots, k + m \) for some \( k \) and \( m \).

A \( u \rightarrow v \) transfer is a transfer of the second type if the labels of the end-vertices of the transferred end-edges form two sections \( k, k + 1, \ldots, k + m \) and \( l, l + 1, \ldots, l + m \) for some \( k, l, m \).

Transfers of the first type work according to the lemma above if \( f(u) + f(v) = k + (k + m)(= k + 1 + (k + m - 1) = k + 2 + (k + m - 2) = \cdots) \). Transfers of the second type work if \( f(u) + f(v) = k + l + m (= k + 1 + (l + m - 1) = k + 2 + (l + m - 2) = \cdots) \).

**Theorem.** Every tree of diameter 4 is graceful.

**Proof.** (as appears in [HH01], although it was first stated by [Zha89]) It is sufficient, using the above lemma, to prove that every tree \( T \) of diameter 4 having the central vertex of an odd degree has a graceful labeling such that the label of the central vertex is maximal.

Let \( x \) be the number of vertices of an even degree that are adjacent to the central vertex of \( T \).

Let \( y \) be the number of vertices of odd degree greater than 1 that are adjacent to the central vertex of \( T \). Let the degree of the central vertex of \( T \) be \( 2k + 1 \) and let \( T \) have \( n \) edges. We can
obtain a graceful labeling of $T$ starting with the tree on the right in the Figure 11 by carrying out the following transfers:

$$0 \rightarrow n - 1 \rightarrow 1 \rightarrow n - 2 \rightarrow 2 \rightarrow n - 3 \rightarrow \cdots,$$

where the first $x$ transfers are of the first type and the next $y - 1$ (if $y > 1$) transfers are of the second type (to get the desirable sets of end-edges of even cardinality).

We are now going to look at trees of diameter 5. The proof given by Hrnciar and Haviar in [HH01] is a bit too technical and involve a large number of cases, so, we are only going to give a sketch here. The authors first show using the above methods that every tree with diameter 5 is “nearly” graceful and then they prove the main result.

So, let $T$ be a tree of diameter 5. Then it has two central vertices which we denote by $a$ and $b$. Let $x$ be a vertex adjacent to the central vertex $a$ such that $x \neq b$. The subtree $T_{a,x}$ is a branch (at the vertex $a$) if $T_{a,x}$ is a subtree of diameter 2. A branch $T_{a,x}$ is an odd branch if the degree of the vertex $x$ is even, otherwise, $T_{a,x}$ is an even branch. Similarly, we define even and odd branches $T_{b,y}$ adjacent to $b$.

Now, let $p = \#odd \ branches \ at \ a$, $r = \#even \ branches \ at \ a$, and $i = \#endedges \ at \ a$. Similarly, let $q = \#odd \ branches \ at \ b$, $s = \#even \ branches \ at \ b$, and $j = \#endedges \ at \ b$.

The graceful labelings defined in the sequel depend on those cardinalities, mostly on their parties. In fact, the authors introduced the following notation: for example $(p, r, i; q, s, j) \equiv (e, o, o; e, e, e)$ if $p, q, s, j$ are even and $r, i$ are odd.

**Theorem.** Every tree $T$ of diameter 5 is graceful or nearly graceful, i.e. if the cardinality of its edge set is $n$, then, there exists a vertex labeling with the numbers from 1 to $n$ such that such that the cardinality of the induced edge labeling is either $n - 1$ or $n - 2$, i.e. at most 2 edges have the same label.

**Proof.** The proof of this theorem looks at a number of cases and exhibits specific transfers which give graceful or nearly graceful labelings of the given tree. One can find the proof of this theorem in [HH01]. We omit it here since it is very long and technical.

Now, the main theorem of this section is stated as follows.

**Theorem.** Every tree of diameter 5 is graceful.
**Proof.** Similarly as in the proof of the previous theorem, the authors distinguish a number of different cases, and using the already found graceful or nearly graceful labelings of the trees, they produce graceful labelings via the transfers defined in the beginning of the section.

**Trees having an even or quasi even degree sequence**

In 2007, Balbuena et al. [BGMV07] defined trees having an even or quasi even sequence to prove them to be graceful. Using a formation algorithm, this method is able to form infinite graceful trees following some rules. Here we show how they proved it and the procedure of formation.

Let \( T \) be any tree and let us denote the degree of any vertex \( u \in V(T) \) by \( \delta_T(u) \) and its neighboring by \( N_T(u) \). We also use \( d_T(u, v) \) to denote the distance in \( T \) between any two vertices \( u \) and \( v \). If \( T \) has even diameter \( D \) and it is rooted in its central vertex \( t \), then \( V(T) = \cup_{1 \leq i \leq D/2} L_i \cup \{ t \} \) where \( L_i = \{ v \in V(T); d_T(v, t) = i \} \). If \( T \) has odd diameter \( D \) and it is rooted in its two (adjacent) central vertices \( a, b \), then \( V(T) = \cup_{1 \leq i \leq (D-1)/2} L_i \cup \{ a, b \} \) where \( L_i = \{ v \in V(T); d_T(v, a) = i, d_T(v, b) = i + 1 \} \cup \{ v \in V(T); d_T(v, b) = i, d_T(v, a) = i + 1 \} \).

In both cases the sets \( L_i \) are called the levels of the rooted tree.

*Definition.* A rooted tree \( T \) with diameter \( D(T) \) has an **even degree sequence** if every vertex has even degree except for one root and the leaves, which are in level \( L_{[D(T)/2]} \). Analogously, a rooted tree \( T \) with diameter \( D(T) \) is said to have a **quasi even degree sequence** if every vertex has even degree except for one root, the vertices in level \( L_{[D(T)/2]} \) and the leaves, which are in level \( L_{[D(T)/2]} \). See Fig. 11 as an example.

![Figure 12. Drawing of a tree having a quasi even degree sequence](image.png)

In [BGMV07] is proved that every tree having an even or quasi even degree sequence is graceful. To do that they find for a tree of even diameter and rooted in its central vertex \( t \) of
degree $\delta(t)$ up to $\delta(t)!$ graceful labelings if the tree has an even or quasi even degree sequence.

First, the authors obtain a relationship between the parities of the number of edges $m$ and the diameter of a tree $T$ having either an even degree sequence or a quasi even degree sequence.

**Lemma.** Let $T$ be a tree of $m$ edges and even diameter $D$. If $T$ has an even degree sequence then $m + D/2$ is even. If $T$ has a quasi even degree sequence then $m + D/2$ is odd.

**Proof.** By definition of even or quasi even degree sequence the root $t$ has odd degree $\delta_T(t)$, hence $|L1| = \delta_T(t)$ is odd. As for all $i = 2, \ldots, \frac{D}{2} - 1$, we have

$$|Li| = \sum_{w \in L_{i-1}} (\delta_T(w) - 1) = \sum_{w \in L_{i-1}} \delta_T(w) - |L_i - 1|$$

and $\delta_T(w)$ is even, then both numbers $|L_i - 1|$ and $|L_i|$ must have the same parity. Therefore $|L1|, \ldots, |L_D/2 - 1|$ are odd numbers. Thus, if $T$ has an even degree sequence, then $|L_{D/2}|$ is also odd and therefore both $m = \sum_{i=1}^{D/2} |L_i|$ and $D/2$ have the same parity. Finally if $T$ has a quasi even degree sequence, then $|L_{D/2}| = \sum_{w \in L_{D/2}} (\delta_T(w) - 1)$ is even because $\delta_T(w)$ is now odd, and hence $m$ and $D/2$ have different parity.

The authors of [BGMV07] follow the terminology introduced by Hrnčiar And Haviari [HH01]. More precisely, let $T$ be a tree and let $uu_1 \in E(T)$. By $T_{u,u_1}$ it is denoted the subtree of $T$ induced by the set of vertices $V(T_{u,u_1}) = \{w \in V(T): w = u$ or $u_1$ is on the $u - w$ path).

Moreover they use the method introduced in Lemma 3 of [HH01] to transform a graceful tree $T$ into a new graceful tree $H$ of the same size. This new graceful tree $H$ is obtained from $T$ by shifting pairs of subtrees $T_{u,u_1}$, $T_{u,u_2}$ from vertex $u$ to some vertex $v \in V(T)(V(T_{u,u_1}) \cup V(T_{u,u_2}))$ such that $f(u) + f(v) = f(u_1) + f(u_2)$, a transfer that we will call a pair transfer. Or by transferring one subtree $T_{u,u_1}$ from vertex $u$ to vertex $v \in V(T) \setminus V(T_{u,u_1})$ if $2f(u_1) = f(u) + f(v)$, a transfer that is called a single transfer. Clearly if the unique $u - v$ path in the tree does not contain a neighbor $u_i$ of $u$ then $v \in V(T_{u,u_1})$. This is guaranteed if $uu_1$ is an end-edge in the tree or if $uv$ is an end-edge in the tree. In order to illustrate this method see Figure 13 in which they perform first a single transfer $5 \rightarrow 1$ attaching to 1 the subtree $T_{5,3}$ because $5 + 1 = 2 \cdot 3$. And a pair transfer $5 \rightarrow 0$ attaching to 0 the subtrees $T_{5,1}$ and $T_{5,4}$ because $5 + 0 = 1 + 4$. 

29
By using the above procedure every tree of diameter four or five was shown to have a graceful labeling [HH01].

Following the same lines of reasoning they prove that it is possible to obtain up to $\delta_T(t)!$ graceful labelings for a tree of even diameter $D$, rooted in its central vertex $t$, and having an even or quasi even degree sequence.

**Theorem.** Let $T$ be a tree of even diameter $D$ and rooted in its central vertex $t$. Consider the set of $\delta_T(t)$ labels

$$X = \{0, 1, ..., \frac{\delta_T(t) - 1}{2}, m - \frac{\delta_T(t) - 1}{2}, ..., m - 1\}$$

Suppose that $T$ has either an even degree sequence or a quasi even degree sequence. Then any numbering of the vertex set $N_T(t)$ with labels in $X$ can be extended to a graceful labeling $f$ on $V(T)$ such that $f(t) = m$.

**Proof.** Notice that the root $t$ has odd degree $\delta_T(t) \geq 3$ because $t$ is a central vertex and because of the definition of even or quasi even degree sequence. Let $k = (\delta_T(t) - 1)/2$. Start by assigning $f(t) = m$ and distributing the set of labels $X = \{0, 1, 2, ..., k, m - k, m - k + 1, ..., m - 1\}$ among the neighbors of $t$ in any order. If $D = 2$ then it is done. So assume $D \geq 4$. A graceful labeling of $T$ is obtained by constructing a sequence of graceful trees of the same size as $T$

$$H^0, H^{m-1}, H^1, H^{m-2}, H^2, H^{m-3}, ..., T$$

using transfers of end-edges. As 0 has been assigned to one neighbor of the root $t$ the value of $\delta_T(0)$ is known. Two cases are studied according to the degree of 0 in $T$. 

---

**Figure 13. Example of a simple transfer and a pair transfer**
Case $\delta_T(0)$ even. The starting graceful tree $H^0$ is shown in Figure 14 in which the children of 0 form the set $\{k + 1, k + 2, \ldots, m - k - 1\}$. Notice that except for $m - k - 1$ all the elements of this list can be associated in pairs whose sum is $m - 1$, i.e., $(k + 1) + (m - k - 2) = (k + 2) + (m - k - 3) = \cdots = m - 1$. Thus, denoting by $k_0 = \delta_T(0)/2$, the graceful tree $H^{m-1}$ is obtained from $H^0$ by successive transfers $0 \rightarrow m - 1$ shifting the subtrees $H^0_{0,k+k_0}$ and $H^0_{0,m-1-k-k_0}$, the subtrees $H^0_{0,k+k_0+1}$ and $H^0_{0,m-2-k-k_0}$, etc, such that the children of $m - 1$ in $H^{m-1}$ form the set of consecutive labels $\{k + k_0, \ldots, m - (k + k_0) - 1\}$ and hence $\delta_H^{m-1}(0) = \delta_T(0)$, since 0 remains adjacent to $\{k + 1, \ldots, k + k_0 - 1\} \cup \{m - (k + k_0), \ldots, m - k - 1\}$. Now, we label the neighbors of 0 in level 2 of $T$ as they have been labeled in $H^{m-1}$.

Similarly, for the second step in the sequence, except for $k + k_0$ all the elements of the current list of children in $H^{m-1}$ of $m - 1$ can be associated in pairs whose sum is $m$, i.e., $(k + k_0 + 1) + (m - 1 - k - k_0) = (k + k_0 + 2) + (m - 2 - k - k_0) = \cdots = m$. So denoting by $k_{m-1} = \delta_T(m - 1)/2$, the graceful tree $H^1$ is obtained from $H^{m-1}$ by successive transfers $m - 1 \rightarrow 1$ such that the children of 1 in $H^1$ form the set of consecutive labels $\{k + k_0 + k_{m-1}, \ldots, m - (k + k_0 + k_{m-1})\}$ which implies that the remaining children of $m - 1$ in $H^1$ are $\{k + k_0, \ldots, k + k_0 + k_{m-1} - 1\} \cup \{m - (k + k_0 + k_{m-1} - 1), \ldots, m - (k + k_0) - 1\}$. Then $\delta_H^1(m - 1) = \delta_T(m - 1)$. Next the children of $m - 1$ are labeled in $T$ as they have been labeled in $H^1$.

This procedure continues by constructing $H^\beta$ from the preceding $H^\alpha$ as long as $\delta_T(\alpha)$ is even (stopping otherwise), by successive pair transfers (and perhaps one single transfer) $\alpha \rightarrow \beta$ in such a way that:

• The children of $\beta$ in $H^\beta$ form a set of consecutive labels.

• $\delta_H^\beta(\alpha) = \delta_T(\alpha)$.

• The children of $\alpha$ are labeled in $T$ as they have been labeled in $H^\beta$. 

Figure 14. Starting graceful tree
If $T$ has an even degree sequence then this procedure stops when all the vertices of $T$ have been labeled and so we are done. So assume that $T$ has a quasi even degree sequence, then the above procedure finishes in a graceful tree $H^{α_0}$ such that $δ_T(α_0)$ is odd with $α_0$ belonging to the level $L_{D/2−1}$ of $T$. The children of $α_0$ in $H^{α_0}$ form a consecutive list of labels. Let $H^{β_0}$ be the successor tree. If $m$ is even then $D/2 − 1$ is even by the Lemma stated before, which implies that $α_0 + β_0 = m$, because in this case all the levels $L_j$ with $j$ even start with a transfer $w → w'$ such that $w + w' = m$. Similarly if $m$ is odd then $D/2 − 1$ is odd by the same Lemma, which implies that $α_0 + β_0 = m − 1$, because in this case all the levels $L_j$ with $j$ odd start with a transfer $w → w'$ such that $w + w' = m − 1$. Now construct the successor tree $H^{β_0}$ by successive pair transfers $α_0 → β_0$ in such a way that $α_0$ remains adjacent to its first child if $m$ is even, or its last child if $m$ is odd, remains adjacent to vertex $[m/2] = (α_0 + β_0)/2$ and also remains adjacent to all the necessary pairs until completing its odd degree in $T$. After this operation the children of $β_0$ in $H^{β_0}$ form two disjoint sets of consecutive labels, $δ_{H^{β_0}}(α_0) = δ_T(α_0)$ and now the children of $α_0$ are labeled in $T$ as they have been labeled in $H^{β_0}$. We continue in this way by constructing $H^β$ from the preceding $H^α$ following this second procedure:

- The children of $β$ in $H^β$ form two disjoint sets of consecutive labels.
- $δ_{H^β}(α) = δ_T(α)$.
- The children of $α$ are labeled in $T$ as they have been labeled in $H^β$. 

![Figure 15. Process of labeling of a tree having a quasi even degree sequence](image)

![Figure 16. Labeled tree having a quasi even degree sequence](image)
To illustrate these procedures let us consider the tree Figure 12. First of all we assign to the root the label $m = 26$ and distribute among its neighbors the labels 0, 25, 1 so we have $\delta_T(0) = 4$. The starting tree $H_0$ is in Figure 14 in which the children of 0 are all the non-distributed labels. By applying the first procedure we obtain the tree of diameter 5 in Figure 15, in which $\alpha_0 = 24$ and $\beta_0 = 2$. Then vertex 24 can give to vertex 2 two sets of consecutive labels, from 6 to 12 and from 14 to 20, in such a way that 13 and 5 remains adjacent to vertex 24. Now procedure 2 continues by transferring from 2 to 23 all the labels except labels 12 and 20, that is, the children of 23 are 6, 7, 8, 9, 10, 11 and 14, 15, 16, 17, 18, 19; 23 transfers to 3 all the labels except 6 and 14; 3 transfers to 22 all the labels except 11, 19, 10 and 15; 22 gives to 4 all the labels except 7 and 16, and finally 4 gives to 21 labels 8 and 17. The resulting tree is in Figure 16.

![Figure 17. Different labeling for the same tree](image)

Another graceful labeling for this tree is in Figure 17 in which we have started by distributing the labels 0, 1, 25 among the neighbors of 26 in a different way.

Case $\delta_T(0)$ odd . Then $T$ has a quasi even degree sequence and $1 = D/2 = 1$, i.e. $D = 4$. Proceed following the second procedure defined in the above case until finishing.

**Theorem.** Let $H$ be a tree of odd diameter $D$, $m$ edges and having an even degree sequence or a quasi even degree sequence. Then $H$ is graceful.

**Proof.** Let $a$ and $b$ be the two adjacent roots of $H$. By definition of even or quasi even degree sequence one root must have odd degree and the other one even degree. Without loss of generality it can supposed that $\delta_H(a)$ is odd, and denote $N_H(a) = b = \{a_1, a_2, ..., a_{\delta_H(a) - 1}\}$. Consider the tree $T$ obtained from $H$ by identifying $a$ and $b$, say $a = b = t$, hence the degree of $t$ is $\delta_T(t) = \delta_H(a) + \delta_H(b) - 2$ which is an odd number, the diameter of $T$ is $D - 1$ even, and $T$ has an even degree sequence or a quasi even degree sequence. Thus by applying their former Theorem, we can define a graceful labeling $f$ of $T$ satisfying that $f(t) = m - 1$ and
\[ f(N_T(t)) = \{0, 1, \ldots, \frac{\delta_T(t) - 1}{2}, m - 1 - \frac{\delta_T(t) - 1}{2}, \ldots, m - 2\}, \]

in such a way that the labels of the set \( N_H(a) - b \) are
\[ f(N_H(a) - b) = \{f(a_1), f(a_2), \ldots, f(a_{\delta_H(a) - 1})\} \subseteq \{0, m - 2, 1, m - 3, \ldots\}. \]

That is, the labels of \( N_H(a) - b \) can be ordered by pairs \((f(a_i), f(a_j))\) satisfying \( f(a_i) + f(a_j) = m - 2 \). Now let \( T' \) be a tree obtained from \( T \) by joining one new vertex \( t' \) and attaching to vertex \( t \) the pendant edge \( tt' \), hence \( \delta_{T'}(t) = \delta_T(t) + 1 \) is even and \( |V(T')| = |V(H)| \). Define a labeling \( \lambda \) on \( V(T') \) such that \( \lambda(t') = 0 \) and \( \lambda(h) = f(h) + 1 \) if \( h \in V(T) \). So \( T' \) is a graceful tree of \( m \) edges and the neighbors of central vertex \( t \), labeled \( \lambda(t) = m \), satisfy that
\[ u \in N_{T'}(t) \text{ if } \lambda(u) \in \{0, 1, \ldots, \frac{\delta_{T'}(t)}{2}, m - \frac{\delta_{T'}(t)}{2}, \ldots, m - 1\}. \]

Then the labels of \( N_H(a) - b \) can be ordered by pairs satisfying \( \lambda(a_i) + \lambda(a_j) = m \). Thus a graceful labeling of \( H \) is obtained from the labeling \( \lambda \) of \( T' \) by shifting \((\delta_H(a) - 1)/2\) pairs of subtrees \( T'_{t,a_i} \) and \( T'_{t,a_j} \) for \( a_i, a_j \in N_H(a) - b \) such that \( \lambda(a_i) + \lambda(a_j) = m \) from vertex \( t \) to the new vertex \( t' \) labeled with 0. By the way of illustration see Figure 18.

![Figure 18. Process to label a tree having an even degree sequence and odd diameter](image-url)
3.3. Computational algorithms

Trees having 27 or less vertices

Another approach to unraveling the graceful tree conjecture is by proving that all trees with up to a certain number of vertices are graceful (or finding a counterexample) with the aid of computers. It was first shown by Alfred and McKay [AM98] that all trees with up to 27 vertices are graceful.

Trees with up to 35 vertices are graceful

This result was extended to 29 in 2003 by Horton [Hor03], who used a randomized backtracking search for graceful labelings. Motivated by Horton’s work, in 2010 Fang used a deterministic backtracking algorithm to prove that all trees with at most 35 vertices are graceful [Fan01].

The algorithm that Fang uses is a hybrid algorithm consisting of two parts, a back-tracking deterministic search and a hill-climbing tabu search combined with some idea from simulated annealing. Here is a description as it appears in the original paper. Suppose we have a tree $T$ on $n$ vertices. The first part is the deterministic backtracking search which tries to construct a graceful labeling $f$ with $f(r) = 1$, where $r$ is the root. This is done by assigning values to vertices one by one. At each recursive call, it tries to make sure that a new value $k$ appears in the induced labeling $g$. The value $k$ decreases as we go deeper in the decision tree, from $n$ to 2. This mechanism assures correctness of this algorithm.

To assure that a new value $k$ appears in the range of $g$, it finds a not-yet-assigned vertex $v$ connected to another vertex $v'$ that is already assigned a label $f(v')$, then tries to assign to $v$ a not-yet-assigned label $f(v)$ such that $|f(v') - f(v)| = k$. There may be several possibilities, or none. If this attempt fails, it tracks back, restores its status and pursues another possibility if there exists one.

Since the decision tree can grow exponentially in size when $n$ increases, we manually add a threshold on the number of backtrack. This prevents searching for a very long time.

This threshold is tuned with respect to the performance of the probabilistic search described. It is empirically fixed to $11000(n - 19) - 1000$ in this verification. For a new, improved version of probabilistic search, it is empirically fixed to $1000(n - 18)$. A detailed version of the
algorithm can be found in the original paper [Fan01].

The second part which is a probabilistic search has the goal of minimizing the following evaluation function of a labeling $f$: $Eval(f) = \sum_{i \in \{1, \ldots, n-1\}} |f(x) - f(y)|: \{x, y\} \in E$. Then, $Eval(f)$ is positive if $f$ is not a graceful labeling and 0 if $f$ is a graceful labeling. By minimizing $Eval$, we can efficiently explore labelings that are likely to be graceful.

The algorithm uses hill-climbing (in this case, hill-descending) to try to minimize Eval. At each iteration, the algorithm tries a number of random modifications and picks the one with the best evaluation. This number is fixed to $2n$.

However, it is known that the hill-climbing method can be trapped in a local minimum. In order to avoid this problem, we use a tabu search. The algorithm keeps track of a number of previous modifications and forbids such modifications unless the result is a graceful labeling. Therefore, the algorithm always goes forth to search for new solutions.

The number of forbidden previous modifications is fixed to be $[n/3]$. This value is determined empirically.

Also in order to solve the local minimum problem, the algorithm accepts with a certain probability determined empirically, modifications that worsen the solution. This behavior is intended to emulate simulated annealing, which can escape local minimum with a probability determined by its “temperature”.

A detailed pseudocode of this part of the algorithm can be found in the original paper [Fan01]. The hybrid algorithm is a combination of the two parts described above. In the first stage the algorithm runs the deterministic backtracking search. If it fails to find a graceful labeling, it turns to the second stage, where it performs the probabilistic search.

The reason for this strategy is that the deterministic backtracking outperforms the probabilistic search in most cases, but in some cases it takes an enormous amount of time. The probabilistic search is not fast compared to the deterministic one, but its runtime varies much less. Therefore, it is natural to use the deterministic search with a cutoff of runtime, then patch unfinished cases with the probabilistic search.

By applying the hybrid algorithm to every tree with at most 35 vertices, Fang verified that every such tree is graceful. Various statistics for the algorithm are given in detail in the original paper.
Here is presented the algorithm in pseudocode used by Aldred and McKay in [AM98] to verify that every tree containing 27 vertices or less admits a graceful labeling, and every tree containing 26 vertices or less admits an harmonious labeling.

For a given tree $T$ and labeling $L$ of the vertices, let $z(T, L)$ be the number of distinct edge labels.

For $n = |V(T)|$, the aim is to find $L$ such that $z(T, L) = n - 1$.

If $L$ is a labeling and $v, w \in V(T)$, define $S_{vw}(L; v, w)$ to be the labeling got from $L$ by swapping the labels on $v$ and $w$.

The method is like this, using a parameter $M$:

1. Start with any labeling of $V(T)$.
2. If $z(T, L) = n - 1$, stop.
3. For each pair $\{v, w\}$, replace $L$ by $L' = S_{vw}(L; v, w)$ if $z(T, L') > z(T, L)$.
4. If step 3 finishes with $L$ unchanged, replace $L$ by $S_{vw}(L; v, w)$, where $\{v, w\}$ is chosen at random from the set of all $\{v, w\}$ such that
   (a) $\{v, w\}$ has not been chosen during the most recent $M$ times this step has been executed.
   (b) $S_{vw}(L; v, w)$ is maximal subject to (a).
5. Repeat from step 2.

This method can be described as a combination of hill-climbing and tabu search. Sometimes it appears to "get stuck" and needs to be restarted from step 1 with a new labeling chosen at random.

A value of $M = 10$ seems ok for small trees, but a slightly larger value seems to be needed for larger trees. The purpose of $M$ is to prevent the algorithm from repeatedly cycling around within some small set of labelings.

Since the generation algorithm produces trees in an order such that most trees are very similar to the previous tree, it proved advantageous to use the graceful/harmonious labeling of each tree as the starting point for the next tree.
4. Adjacency Matrices

4.1. Description and representation

An adjacency matrix $A$ of some graph $G (V, E)$ ($n$ vertices) gets represented by a $n \times n$ matrix such that $a_{ij} = 1$ if the edge $(i, j)$ belongs to $E$, and $a_{ij} = 0$ if it does not (there is no such edge in our graph), $i, j \in V$. Working with matrices instead of graphic representations will let us detect graph properties faster in exchange for the instinctive and global view that graphic representation provides us.

In this paper we will represent binary values in adjacency matrices as dots instead of ones and voids instead of zeroes.

4.2. General properties

An adjacency matrix of a graph stores all of its information. Basic properties of a graph such as number of edges or maximum or minimum degree, etc. are directly shown on its matrix. An example of a matricial representation of a graph is shown on Figure 19.

![Figure 19. Matricial representation for a graph](image)

Given that we will only consider bidirectional graphs, every adjacency matrix shown will be symmetrical, as $a_{ij} = a_{ji}$. The total amount of dots represented in the matrix will be $2m$, $m$ in
each triangle, \( m \) representing the total amount of edges in \( E \). In order to reduce the complexity of the matrix representation, only the lower triangle will be represented.

If a graph is complete, its matrix will be full excluding the diagonal, always void, and if the graph is connected, it will be possible to move from any dot in its adjacency matrix to any other using vertical or horizontal movements between dots.

**Proposition.** Rows and columns shall be ordered obeying the order of some labeling of the vertices \( f: V(G) \to \{0, 1, \ldots, m\} \).

The number of dots in a row or column indicates the number of neighbours the vertex represented by that row or column has, its degree. If only the lower triangle it is represented, this number it is obtained adding the dots in the part of the row and the column included in the lower triangle.

If dots in \( A \) represent the edges in the graph, it is because the matrix is telling us there is a path of distance one between the two vertices of the edge. Different powers of the matrix give us different path lengths. So, if \((i,j)\) in \( A^x \) is not void, this means there is at least one path of length \( x \) between vertices \( i \) and \( j \). This property does not tell apart paths that step on the same vertex more than once.

### 4.3. Adjacency matrices for trees

Any \( n \times n \) adjacency matrix representing a tree contains \( 2m \) dots, being \( m \) its number of edges, with \( m = n - 1 \), \( m \) dots in each triangle. A matrix containing \( m \) dots in each triangle will represent a tree if and only if it contains no cycles.

To find a cycle in an adjacency matrix we have to realize that when a row or a column contains more than one dot that represents more than one edge attached to the same vertex. Therefore any path that can be made jumping vertically or horizontally between dots will be a real path in the graph. If there exists more than one path that connects two different dots in the matrix then we can say that the matrix contains a cycle, so it cannot represent a tree.

This property has a remarkable importance, given that while a cycle is relatively easy to spot in a graphic way, once a graph is coded into an adjacency matrix it can get significantly harder. If we only consider the lower triangle of the matrix, paths from dot to dot can use the positions in the diagonal matrix as middle steps between one dot and another.
4.4. Labeling changes

Any labeling change is easy to represent in an adjacency matrix as it gets represented as a row swapping (and following column swapping, since it is symmetrical) guided by the labeling change. Algebraically this can be represented with a row swapping matrix $P$, which is the $n \times n$ identity matrix with the required rows swapped. Resulting new labeling matrix will be $A' = P \cdot A \cdot P^T$. An example is produced in Figure 20.

![Image: Matricial representation of a labeling change on a tree](image)

Figure 20. Matricial representation of a labeling change on a tree
5. Graceful adjacency matrices

5.1. Matricial representation for graceful labelings

Definition. We define oblique line at distance $d$, $0 < d < n$, as the set of matrix values $(i, i - d)$ for every $i, 0 < i < n$ in an adjacency matrix, representing this way each value line parallel to the main diagonal if the matrix, each one of them represented by a different distance $d$.

Each edge included in the set represented by a unique oblique line at distance $d$ is bound to be assigned by the graph’s labeling an induced value $i - (i - d) = d$, and in the other hand each oblique line represents every edge with an induced value of $d$.

Figure 21. Matricial representation of a graceful labeling on a tree
Theorem: A tree is gracefully labeled if and only if its adjacency matrix, ordered considered such labeling, presents exactly one dot in each one of the oblique lines included in its lower triangle. Analogously, the same can be said on the upper triangle.

Proof: If there existed an oblique line at distance $d$ without any one on it that would imply an absence of edges with an induced label of $d$, and given that $0 < d < n$, the labeling would not be graceful. If there exists an oblique line at distance $d$ with more than one one in it, the labeling will induce the value $d$ on more than one edge, making it impossible for the labeling to be graceful. In Figure 21 a matrix representing a graceful labeling on a tree is shown.

This theorem, although initially genuine, as I found out later, had already been published by Charles Michael Cavalier in [Cav06].

Definition: We will call Graceful adjacency matrix any matrix containing one and only one one in each of its oblique lines, leaving the main diagonal empty. The remains of this thesis are headed to investigate the possibilities offered by these matrices.

To begin with we will reformulate Rosa’s conjecture: “Every tree is graceful” can be stated equivalently regarding graceful adjacency matrices:

Conjecture: For any $A$ adjacency matrix of a tree $T$ there exists some $P$ row swapping matrix such that $A' = P \cdot A \cdot P^T$ becomes a graceful adjacency matrix.

Inverted trees

![Inverted tree on adjacency matrix](image)
An easy way to know if our adjacency matrix has a cycle is to draw the inverted tree between the dots of our matrix, as shown on Figure 22. Further research on properties of inverted trees as defined here can be made in some future.

5.2. Adjacency matrices for gracefully proven tree classes

In this section we will analyse the different graceful adjacency matrices produced when representing classes of trees already proven to be graceful.

Stars, paths and caterpillars

As we can see in the drawings displayed in Figure 23, stars will be always represented by a full first column or last row (main diagonal left empty), given that the only way to label a star gracefully is assigning the lower or the higher value to its central vertex.

Paths, on the other hand, show a zig-zag path through its graceful adjacency matrix like the one we can see in Figure 23, when they are labeled like caterpillars. As before, it is possible to represent the adjacency matrix of the inverse labeling by applying a symmetry with the perpendicular to the main diagonal as axis.
Paths and stars are studied because they represent the basic components of every other tree. On the other hand, every tree is also decomposable into caterpillars.

Caterpillars, when labeled as in [Ros67], the most common and easy way to label them, present a stairlike shape in its adjacency matrix, changing direction in every vertex of its central path. This stairlike shape might be interpreted as a conjunction of stars in a path, which is essentially what a caterpillar is. Figure 24 represents the graceful adjacency matrix of a caterpillar on 24 vertices.

![Figure 24. Matricial representation of a graceful caterpillar](image)

Most of other gracefully known classes that resemble caterpillars, such as fireworks, spraying pipes, or even some spider like trees such as symmetrical trees or olive trees, present a similar stairlike shape on its adjacency matrix, with the only difference that certain corner dots are
displaced along their oblique line. A small change like this one can completely change the appearance and class of a tree.

As it is easily seen, none of these three classes of adjacency matrices, stars, paths, or caterpillars, present dots, considering their lower triangle, in an upper or more to the right position than the one closest to the main diagonal. As is it shown later, this means that their graceful labelings are also bipartite labelings ($\alpha - valuation$, [Ros67]). This property will be very important in the chapters ahead.

**Spiders**

Spiders, when labeled as in Section 3.3., present a graceful adjacency matrix with two diagonals of dots alternating oblique lines. See Figure 25 for a graphical explanation.

![Figure 25. Matricial representation of a graceful spider with five legs](image)

**Trees having an even or a quasi-even degree sequence**

The process described in [BGMV07] of transferring all the vertices with the middle labels from one vertex to another is easily trackable on a graceful adjacency matrix, as represented on Figure 26. Notice that in the second matrix in Figure 26, dots represent transitory edges, whereas squares represent the final ones.
Figure 26. Matricial representation of a process of formation of a graceful tree having an even degree sequence and odd diameter
5.3. Coding

We will now provide a codification method for graceful adjacency matrices that is not valid for other kinds of matrices. Although this method was first thought to be genuine, later I found that it had already been published, with some irrelevant differences.

Given a graceful adjacency matrix of a graph $G$ with $V = 0, ..., n - 1$, we consider every one’s position in the lower triangle as $u_{i,j}, i - j = d$. Said matrix can also be expressed as a vector $v(d) = j$, for every $d$, given the fact that every $d$ identifies with one and only one edge. Furthermore, $v(n - 1) = 0$, because the oblique line at distance $n - 1$ includes the value $u_{n,1}$ as its only value, which is always one, so $v(n - 1)$ becomes an irrelevant value and can be excluded from $v$. Similar thing happens with $v(n - 2)$, given that we can set the value on 0, and if our matrix has $v(n - 2) = 1$, we can apply a labeling inversion (symmetry with perpendicular to diagonal as axis) and the value will become 0 and the matrix will keep representing the same tree. So $v(n - 2)$ can be also excluded from $v$. Finally we obtain a vector with $n - 3$ components, whereas normally to represent a tree $2n - 2$ components are needed ($n$ vectors, 2 components each), and the full matrix would normally take $n^2$ components to be represented.

This code had already been stated as $v(x) = \{a | (a, b) \in E, b > a, b - a = x\}$. Observe that this formulation would return the same values as the stated above, but in inverse order.

This kind of codification can be useful in cryptology fields, the use of which is far extended nowadays.

**Example**

To give an example, let us consider the tree shown in Figure 27, already gracefully labeled and with its graceful adjacency matrix also represented.
The classic code for a graph would be

\[ E = \{(0,5), (0,7), (0,9), (1,9), (2,3), (2,8), (4,7), (4,8), (6,8)\}, \]

which comprises 18 elements. Let us now code it as stated above.

Given that \( n = 2 = 8 \) = 1, we should perform an inversion of the labeling to code it as stated above. Another, and maybe faster, way to do it is to establish \( n = j \) instead of \( n = i \). This way \( n = 0 \) and the final code will be

\[ E' = (6, 1, 2, 1, 4, 1, 2), \]

which comprises 7 elements.
6. Matricial unions for graceful trees

We have already taken a look to the particularities that adjacency matrices have in front of graphical representations of graceful trees. Although Chapter 6 has been developed originally with no external sources needed, the matricial representation of graceful trees has already been researched, as far as I know, in Charles Michael Cavalier’s Bachelor of Science [Cav06]. However, content from now on has never been published, as far as I know. The purpose of the next two chapters is to analyse unions of graceful trees in a way that the resultant tree continues to be graceful. Although Cavalier [Cav06] made an attempt at approaching the matter, there are two mistakes in this approach that make the whole approach void of valid content. Firstly, he considers that given that every tree is bipartite, which is true, every tree’s graceful adjacency matrix will have a bipartite labeling, which is not. Secondly, he creates matrix unions by chaining the whole matrices in a larger matrix, whereas the right way to do it is to chain the lower triangles of the matrices in a larger lower triangle, in order to guarantee the symmetry and to respect the meaning of the matrices, as shown on Figure 28.

Figure 28 Correct positions of graceful tree submatrices for a matricial union
6.1. Bipartite labelings and union requirements

Definition: A labeling is bipartite if there exists some $x$ such that for every edge $(i, j) \in E$, if $i < j$, then $i < x < j$. This means that if we divide all vertices in two classes in a bipartite way, one class' set of labels will be $1, 2, ..., k$ and the other $k + 1, k + 2, ..., n$.

Property: The adjacency matrix representing a bipartite labeling can be divided into 4 submatrices as it is shown in Figure 29, being two of the submatrices empty and the other pair symmetrical with each other.

![Diagram of a bipartite labeling on a graph](image)

Property: Given a graph with a bipartite labeling, if some vertex labeled $i$ has a neighbour labeled $j$ such that $i < j$, then there cannot be a neighbour labeled $k$ of $i$ such that $i > k$. In other words, every vertex has neighbours either with higher labels, or with lower labels, but never both kinds at the same time.

Property: Apart from the already known labeling inversion, bipartite labelings are invertible by rotating its non-empty submatrices 180 degrees, or applying a central symmetry to them. This inversion, combined with the already known, provides us with 3 different possible graceful bipartite labelings if the original one does not suit our needs. Analytically, if we describe the center of symmetry (center of submatrix, lower triangle) as \(\left(\frac{n+a}{2}, \frac{b+1}{2}\right)\), $(a, b)$ being the edge with induced value $1$, then the matricial conversion is:

\[
(i, j) \rightarrow (i', j') = (n + a - i, b - j + 1)
\]
This statement above implies that this inversion affects differently the two classes generated by the bipartite system. Higher values become the rows of the submatrix, and therefore their inversion is $i' = n + a - i$, while lower labels become the columns of the submatrix, and their inversion is given by $j' = b - j + 1$.

Let $T(t)$ and $U(u)$ be two trees with $n$ and $n'$ vertices respectively, already gracefully labeled, with graceful adjacency matrices $A(T)$ and $B(U)$ respectively. We will denote the edge with induced value of one in $T$ as $A$’s “point”, the one closest to the diagonal. We will also denote the edge with induced value $n' - 1$ in $U$ as $B$’s “base”, in the position $(n', 1)$.

The most important thing to bear in mind in this process is that the resulting matrix of the addition of $A(T)$ and $B(U)$, $G$, has to keep being graceful. This means that one matrix’s ones cannot be positioned in oblique lines already occupied by the other matrix’ ones, so matrix $B$ must occupy oblique lines at distances $1, 2, \ldots, n' - 1$ to the main diagonal and $A$ must occupy oblique lines at distances $n' + 1, n' + 1, \ldots, n' + n$ to the main diagonal. This may be a bit different depending on the kind of union that we intend to make.

The result is the integration of matrix $B$ between upper and lower triangle of matrix $A$, as it is shown in Figure 28.

No alteration in the original matrices is considered in this chapter.

**Theorem:** If there is more than one row in which coexist ones from both matrices, then resulting matrix $G$ will not represent a tree.

**Proof:** Let us assume that in our final matrix $G$ there are two rows, $k$ and $l$, with ones from both original matrices $A$ and $B$. From any one in matrix $A$, given that it represents a connected graph, it is possible to move to that one in row $k$ using horizontal or vertical movements between $A$’s ones. From that one we will be able to jump to the one in row $k$ that belongs to $B$. From this one, given that $B$ also represents a connected graph, we will be able to reach the one in row $l$ using horizontal and vertical movements between $B$’s ones. From this one we can jump back to $A$ through the one in row $l$ belonging to $A$, concluding the existence of a cycle. If it is not possible to trace this cycle, then the union has disconnected one of the graphs, and again, $G$ does not represent a tree.

Even though in some cases it is possible in some other way to guarantee the cycle non existence, in a general way we will not position $B$’s base under $A$’s point, or left of it.

Positions studied in this chapter for matrix $B$’s base are matrix $A$’s point $(i,j)$ and adjacent positions closer to the main diagonal: above $(i - 1, j)$, right $(i, j + 1)$, and diagonally above
and right \((i - 1, j + 1)\). Further positions are possible in case of using another tree as a bridge between the first two, anchored to both A's point and B's base (as explained in Section 6.2 or 6.3).

**Theorem:** Assuming that we pretend to anchor B(U)'s base in a position either identical or adjacent (closer to the main diagonal) to A(T)'s point, then if T's graceful labeling is not also bipartite, resulting matrix \(G\) will not represent a tree.

**Proof:** A bipartite labeling represented in a matrix has the quality of dividing the matrix into four submatrices, two of which are void and the other to symmetric to each other. Furthermore, this quality is only achievable by a bipartite labeling. For a graphical understanding see Figure 30.

![Figure 30. Drawing of the position of bipartite submatrix](image)

In other words, a not bipartite labeling implies the existence of at least one dot above or more to the right than the matrix point (lower triangle). This one, assuming it is above the point without loss of generality, will coexist with at least one one from matrix B in its row, since every vertex in U has at least one edge assigned to it. Assuming that a union will be provided without involving changes in other vertices, this one would mean a second union and therefore, a cycle.

In order to execute the union of two gracefully labeled trees, one of them at least having a bipartite labeling, we can distinguish 3 basic ways to do it: Edge identification, vertex identification, and connection
6.2. Edge/vertex identification

**Edge Identification**

This occurs if we position B’s base identified with A’s point. The edge with an induced value of 1 in T and the edge with an induced value of $n’ - 1$ of U become the same edge, and it is new induced value is $n’ - 1$. Vertices included in these edges are also identified. An example is shown on Figure 31.

Labeling changes are as follows: Every label $a$ in A tree higher than its neighbour’s labels become $a’ = a + n’ - 2$. Labels in A tree lower than its adjacents remain the same. In B tree every label $b$ becomes $b’ = b + j - 1$, where $j$ is the column where A’s point is.

![Figure 31. Example of matricial union by edge identification](image)

**Vertex identification**

If B’s base is positioned immediately above A’s point, this means that base’s minor vertex and point’s major vertex are identified. If B’s base is positioned immediately to the right of A’s point, this means that base’s major vertex and point’s minor vertex are identified. If it is needed to cross these possibilities (minor minor, major major), one must invert the labeling of one original tree. An example is shown on Figure 32.

Labeling changes are as follows: In A tree, labels of vertices with minor neighbour labels become $a’ = a + n’ - 1$. Labels in A tree lower than its adjacents remain the same. In B
Every label \( b \) becomes \( b' = b + j - 1 \) if \( B' \)'s base is positioned above \( A' \)'s point, or \( b' = b + j \) if it's positioned to the right, where \( j \) is the column where \( A' \)'s point is.

![Figure 32. Example of matricial union by vertex identification](image)

One of the most known tree addition methods is the one that hangs a caterpillar from a vertex with a zero label. This method is just a specific case of vertex identification.

### 6.3. Connection

Last position possible, \( B' \)'s base is positioned diagonally above and to the right of \( A' \)'s point. A connection edge has to be created in this case, being able to choose any edge in the oblique line between both submatrices, provided that it is an edge with one end in one submatrix and the other end in the other one. If the chosen position's distance from the base/point is greater than \( \min(n, n') \), then the edge is no longer connecting both trees, but two vertices from the same tree. An example is shown on Figure 33.

This method has a great versatility due to being able to choose any vertex as connector. Every vertex has at least 8 connection possibilities in the other tree, provided that every tree's labeling can have one symmetry applied and every bipartite labeling, two symmetries. If both trees have bipartite labelings, then every vertex has 16 potential connections in the other tree, and it is also possible to invert the matrices positions (next
to main diagonal) to enable 16 more potential connections. These potential connections can be reduced by the size of the matrices and by repetition of potential connections.

Labeling changes are as follows: In $A$ tree, labels of vertices with minor neighbour labels become $a' = a + n'$. Labels in $A$ tree lower than its adjacents remain the same. In $B$ tree every label $b$ becomes $b' = b + j$, where $j$ is the column where $A$’s point is. Connection edge induced value is $n'$.

![Diagram](image)

**Figure 33. Example of a matricial union by connection**

### 6.4. Multiple unions

**Successive unions**

In this section the possibility of performing a multiple union is analysed. The ultimate goal of this study is to implement a general method able to gracefully label huge trees by sectioning them into smaller subtrees, preferably subtrees that already have a known graceful labeling method.

Let $A$ be the adjacency matrix for a gracefully labeled tree. Let $B, C, D$ and $E$ be adjacency matrices for gracefully and bipartitely labeled trees. Let $T$ be the tree resulting from the matricial union of $A$ and $B$, $A$ positioned next to main diagonal. Let us call $T'$ to the tree generated by a matricial union of $T$ and $C, T$ positioned next to main diagonal. This is done
successively results in a graceful adjacency matrix representing $T'''$, matricial chained union of $A, B, C, D$ and $E$. Obviously this process can be done with any number of trees. Figure 34 depicts an example of this method.

This method provides us a way to create (or to section) "paths" of trees, gracefully labeled trees arranged serially, forming a large graceful tree. To describe a more general multiple union method, connections need to be changed.

![Figure 34. Representation of a successive union of five submatrices](image)

**Tree of trees**

Let us now assume it is needed to join the same set of matrices but not serially, or maybe the non-bipartitely labeled tree ($A$) has to be placed somewhere else than an end of the "path". If the process described above is followed, but connections are not assigned yet, once every matrix is positioned in the way that connection method describes, the connection possibilities get multiplied, enabling, depending on every submatrix size, direct connection between any two trees.

This process provides us with multiple intermatricial connections, allowing us to choose a configuration of connecting edges that meets our needs. It is necessary to point out that given that every adjacency submatrix can be modified following symmetry methods, or simply assigning an alternate labeling, we must consider the possibility that every tree can
be constructed this way. A drawing representing the connection possibilities for a union of five submatrices is shown on Figure 35.

![Figure 35. Drawing of the possible connections for a union of five submatrices](image)

This method opens the door to being able to label huge trees. Since the computacional complexity of labeling a tree increases in a factorial way with its size, being able to sever the tree into a number of smaller subtrees, give them a labeling, and reunite afterwards, can be a huge time saver. Probably this is also the only way to label trees so huge not even the most powerful computers can solve in a lifetime. To give an example, a 100 nodes tree admits a number of labelings close to $10^{158}$. If we can divide it into 10 subtrees, for example, each one having 10 nodes, finding a graceful labeling for each one of them is trivial.

The easiest way to label a huge tree is to divide it into caterpillars, since caterpillars always admit bipartite labelings, and every tree can be decomposed into them.

**Example**

Let us show an example of the usefulness of this method. Let $T$ be a tree of 50 vertices as shown in Figure 36. As far as we know, $T$ has no special property as symmetry or even or quasi even degree, nor does it belong to any known graceful class of tree. So, as far as we know, there is almost no easy way to gracefully label it, and a computer would probably take ages to find a graceful labeling. Note that a tree on 50 vertices can be labeled in $3 \cdot 10^{64}$ ways, approximately.
We need to find a tree of trees that suit our needs. The first step to do so is to sever it into caterpillars and to label them following the rules stated in Section 3.1., as shown in Figure 37 and 38.
Then it begins the complicate part. We must position these submatrices in the greater matrix in a way that we can connect them as in the original tree and each connection is placed in a unique free oblique line. This step involves considering different sequences of the submatrices and applying symmetries to the submatrices until we find the correct positioning. This can involve a lot of time since there are a lot of different possible positions, but it’s not even close to the time it would take to solve the tree without this method. Finally, we obtain the resulting matrix shown in Figure 39:
The above matrix (Figure 39) gives the final labeling to the tree, shown in Figure 40:
Fractal trees

Let us propose a multiple connection involving $k$ identical subtrees, each one of them has been given a graceful and bipartite labeling, we will denominate each subtree as seed tree. If we can find a tree of trees such that considering each subtree a node, the resulting tree is identical to the subtrees, the we can declare that the repetition of the former procedure using $k$ copies of the obtained tree presents a fractal structure and can be repeated ad infinitum.

Analysing a matrix comprised by identical submatrices arranged diagonally, we can identify a special case. If every submatrix is square, meaning that the two bipartite classes of its labeling have the same number of vertices, and if we number the submatrices as $1, 2, \ldots, k$ from the lower corner to the main diagonal, then connections can only be made between submatrices having different parity number. For example, the lower corner submatrix can only be connected with even numbered submatrices. As there is the same number of odd submatrices as there is of even ones, or a difference of one, the resulting tree of trees has to follow the same rule, as of having the two bipartite classes of vertices the same amount or a difference of one vertex. Notice that in a fractal structure as we described before, this property is automatically fulfilled, as it was a requisite to a square submatrix.
The less “square like” the submatrices are, the easier it will be to connect two submatrix with same parity number.

**Example**

Let us choose the tree shown in Figure 41 to build a fractal structure around it. As we can see it accomplishes having a bipartite graceful labeling, and its tree of trees has also a bipartite labeling, as we can see on Figure 42.

![Figure 41. Seed tree for a fractal structure and its graceful adjacency submatrix](image1)

![Figure 20. Graceful adjacency matrix of the fractal structure](image2)
On Figure 43 it is shown the result for one step of the process. If we consider each subtree just a vertex for the next step, then we can repeat this process ad infinitum.

Figure 43. Seed tree with seed trees as vertices
7. Trees having a perfect matching

7.1 Introduction

The purpose of this chapter is to study the graceful tree conjecture in the context of the trees having a perfect matching, using adjacency matrices to identify their special properties.

A perfect matching is a subset of edges of a graph that includes every vertex of the graph without repeating any. An example is shown on Figure 44. A condition to have a perfect matching is to have an even number of vertices. It is a necessary condition but not sufficient.

![Figure 44. Example of a tree with a perfect matching, which is highlighted](image)

7.2 Strong Gracefulness

Once it comes to assigning a graceful labeling to a tree having a perfect matching, my first attempt was to always consider a labeling of the perfect matching such that every label plus its pair results in \( n - 1 \), where \( n \) is the total number of vertices. This way, every edge included in the perfect matching is given an odd induced value \( n - 2i - 1 \), being \( i \) the lower value of the pair of vertices this edge is connecting. Therefore, every odd value is already covered and every other connection that would induce an even value on its edge can be discarded.

Later on I found out that this had been already described. In fact, graceful labelings of trees having a perfect matching that satisfy the condition described above are known as strong graceful labelings [BH99].
Definition. A tree $T$ on $n$ vertices is strongly graceful if $T$ contains a perfect matching $M$ and $T$ admits a graceful labeling $f$ such that $f(u) + f(v) = n - 1$ for every edge $uv \in M$.

This results in a distribution like the one showed in Figure 45. After this, only even edge values must be assigned. To say it in other words, instead of assigning labels to vertices the problem becomes assigning the perfect matching to the edges.

Note that this could be interpreted as a tree of trees like the ones defined in Section 6.4. This way, every edge would become a subtree and connections would be needed in every even oblique line.

Once this remark has been stated, let us analyse the possible connections between the edges forming the perfect matching. Leaving the odd induced values aside since all of them are covered, Figure 46 shows us the configuration of the connections left possible.
7.3 Reduction of the problem

Once we have established the procedure to find a graceful and strong labeling, there is a way to reduce the complexity of the problem in half.

Definition. The spiketree of a tree $T$ on $n$ vertices is obtained by adding $n$ new vertices to $T$ along with $n$ edges. The contree of a tree $T$ with a perfect matching $M$ is obtained from $T$ by contracting the edges of $M$.

So, it is a consequence of this that if $T'$ is the contree of $T$, then $T$ is a spiketree of $T'$. Note that there is only one contree for every tree with a perfect matching, while there are multiple spiketrees for every tree.

So, if the problem is to find a graceful and strong labeling for a tree with a perfect matching, one possibility is to solve the contree of it, with a new set of rules that guarantee that the spiketree of it that corresponds our initial tree is assigned automatically a graceful labeling.

In Figure 46, one can see that every couple of edges included in the perfect matching can be connected in four ways, identical in two pairs, one of which induces in the connecting edge an odd value. Leaving that possibility aside, since every odd value has been covered in the perfect matching, only two other possible connections are left, which are identical and induce an even value in the connecting edge.

So, there's only one valid induced value to every possible edge connecting two edges from the perfect matching. This means that every two vertices in the contree can be connected and the induced value of its edges follows a bijective function. This bijective function is as follows:

- Let’s label every vertex of the contree with the lowest label of the edge of the perfect matching it represents.
- If an edge is connecting two vertices of the contree labeled $i$ and $j$, having $i$ and $j$ the same parity, then the induced value of this edge becomes $|i - j|$.
- If an edge is connecting two vertices of the contree labeled $i$ and $j$, having $i$ and $j$ different parity, then the induced value of this edge becomes $n - i - j - 1$.

The resulting edge induced values matrix is as follows:
The problem is to place only one dot on each value (maxim value is always an obligatory dot), and to keep the structure of the contree of our tree. Once we have correctly labeled the contree, the following step is to label the original tree.

Every perfect matching pair is labeled according to its representing vertex in the contree, labeled $i$, for example. Then one vertex of the pair is given the same label $i$, and the other one is given the label $n - i - 1$. We begin by labeling one pair, choosing arbitrarily one of the vertices as lower label. Then the next pair’s order will be conditioned to this first decision, since we will have to make sure that only even induced values on edges between pairs are produced. For example, if pair 2 is connected to pair 4 in 45, only possible edges are $(2, 4)$ and $(5, 7)$, since $(2, 5)$ and $(4, 7)$ would induce an odd value on the edge. So, if we label pair 2 first, then if pair 4 is connected to vertex 2, the adjacent vertex will be 4 and if it is connected to vertex 7, the adjacent vertex will be labeled 5.

**7.4 Equivalent conjecture**

Broersma and Hoede [BH99] conjectured on its publication about strong graceful labelings.

*Conjecture. Every tree containing a perfect matching is strongly graceful.*

They also proved in [BH99] that this conjecture is equivalent to the Graceful Trees Conjecture. This is explained in Robeva’s Honors Thesis [Rob11], but I have not been able to find the original paper. In [Rob11], Robeva explains briefly how Broersma and Hoede achieved this
proof, but the original paper would be needed to understand it well, since it is too complicated to be solved in just a paragraph.

Given that I did not have the demonstration of the equivalence, I decided to give the matricial view of the strong gracefulness another sight and ended up proving the conjecture equivalence in a matricial but simple way.

In the attempt to find a simpler rule to label the contree of a tree with a perfect matching, I set out to find a way of converting the induced value matrix as follows:

\[
\begin{array}{cccc}
0 & 8 & & \\
1 & 2 & 6 & \\
2 & 6 & 2 & 4 \\
3 & 4 & 4 & 2 \\
4 & 8 & 6 & 4 & 2 \\
\end{array}
\]

Figure 48. Representation of the necessary edge induced values to be a graceful labeling problem

To obtain this matrix, we can perform a change of variable. Notice that in the example set in Figure 48, an alteration of the order of the rows and columns such as \((0, 1, 2, 3, 4) \rightarrow (0, 2, 4, 3, 1)\) would provide us the ordered matrix we are looking for. The rule this change of variable follows is this one:

- If \(i\) is even, then \(i' = i/2\)
- If \(i\) is odd, then \(i' = n' - (i + 1)/2\)

\[
\begin{array}{cccc}
0 & 1 & & \\
1 & 2 & 1 & \\
2 & 3 & 2 & 1 \\
3 & 4 & 3 & 2 \\
4 & 8 & 6 & 4 & 2 \\
\end{array}
\]

Figure 49. Final edge induced values matrix of the contree after change of variable
Being $i$ the initial label, $i'$ the resulting one from the change of variable, and $n' = n/2$ the number of vertices of the contree ($n$ is the number of vertices of the initial tree).

Finally, we change the induced values to $u' = u/2$ and the resulting induced value matrix becomes the one shown in Figure 49.

Which is exactly an adjacency matrix showing the induced values for a Graceful Labeling problem, and can be solved such as one. Once solved, the change of variable has to be inversed, just as follows:

- $u = 2u'$
- $i = 2i', i < \frac{n'}{2}$
- $i = 2n' - 2i' - 1, i' \geq n'/2$

And afterwards the original tree can be labeled as stated in Section 7.3, proving the equivalence of the two conjectures.

**Example**

Let us show with an example how to label a big tree with a perfect matching using this method. Let $T$ be a tree of 42 vertices like the one shown on Figure 50. $T$ has a perfect matching and it has been already highlighten in the figure. The first step in the process is to obtain its contree, also shown on Figure 50. Afterwards, we give a graceful labeling to the contree of 21 vertices using, for example, a tree of trees like the one shown on Figure 51.
Once the contree is labeled, the next step is to revert the change of variable as explained previously in this chapter in order to find out which edge of the strongly graceful perfect matching is representing each vertex of the contree, as shown on Figure 52. Once this is
known, the initial tree can be labeled, ordering each pair according to have only even differences between vertices of different pairs. Final graceful labeling is shown on Figure 53.

Figure 52. Change of variable of the graceful labeling of the contree

Figure 53. Graceful labeling for a tree on 42 vertices having a perfect matching.
9. Conclusions

The Graceful Tree Conjecture is a thrilling problem. As I said in the introduction, it seems as difficult to solve as it is to enunciate, and the result of this is that you can explain it to anyone in a minute and have this person thinking about it for a week.

Once studied most of the research on this issue, I have come to the conclusion that the approach directions that most of the researchers have taken until now, whereas there is a lot more to explore in that way, don’t seem to offer a possibility of a total solution.

I am not saying that adjacency matrices have a better chance at finding this total solution, but I am still surprised that almost no one has ever tried this approach, at least as far as I know.

In Chapter 6 I have given a method to gracefully label huge trees. As an advantage against every labeling method until this moment, this one does not ask for specifical tree conditions. I conjecture every tree can be labeled this way, regardless of its size, vertex parity, symmetry or diameter. And to remark the size, I believe this method to be the only way to label unclassified trees on more than 50 vertices approx..

The problem with this method is the lack of control on the connectors between subtrees. Further research should be made on this matter to establish a pattern on how these connections work.

On the other hand, in Chapter 7 I developed a whole method to label easily trees having a perfect matching, only to find out weeks later that it had already been stated as strong gracefulness and that it had been proven to be an equivalent to the Graceful Tree Conjecture. However, I developed my own proof about this equivalence, since the original paper was not available. I cannot verify if my proof follows the same lines as the mentioned paper, but I would like to emphasize that my proof is only a few lines long.

I have finished this thesis with the feeling of having really mastered the subject. I have come to know and understand almost every major step that has been done towards the unravelling of this puzzle that is the Graceful Tree Conjecture.

Once again I would like to show my gratitude to my tutor Professor Balbuena who has guided me through this project and offered all the help needed. It has been a pleasure to work with her and learn from her.
8. References


[Cav06] Charles Michael Cavalier, Graceful Labelings, Bachelor of Science (Louisiana State University 2006) http://www.math.sc.edu/~czabarka/Graceful.pdf


