Fibonacci Graphs

Tao Gaede

November 29, 2020

Abstract

This paper was the result of a fun exploration of the following challenge: try to construct families of graphs that contain as many Fibonacci numbers within their structure as possible. To that end, I present constructions of a cycle, and two generalizations of the Fibonacci tree (here called a Fibonacci branch). Since the Fibonacci number sequence is recursive, in every case, the constructed graphs reflect this recursion. A possibly significant result of this paper is the use of a balancing parameter m found in the well known proposition below as a way to inversely relate Fibonacci tree (as defined in this paper) trunk size with the complexity of its branches.

Introduction

We will begin with the Fibonacci numbers and the above mentioned proposition, followed by my construction for a Fibonacci cycle. Fibonacci branches (in the literature called Fibonacci trees) and what I'm calling here Fibonacci trees, and Super Fibonacci trees will be defined in the final section. We will conclude with a discussion on the graph structure implications of the balancing parameter m.

Definition (Fibonacci numbers): The Fibonacci numbers are defined as follows:

$$f_0 = 1, f_1 = 1,$$
$$f_n = f_{n-1} + f_{n-2}$$

The following proposition will be used in the construction of Fibonacci graphs, and is a generalization of the above definition.

Proposition

When $n > m \geq 1$, any Fibonacci number is the sum of two terms, each a product of two smaller Fibonacci numbers.

$$f_n = f_m f_{n-m} + f_{m-1} f_{n-m-1} \tag{1}$$

Proof The proof will proceed by induction on m.

Inductive Hypothesis $P(m) := f_n = f_m f_{n-m} + f_{m-1} f_{n-m-1}$

Base Case Let $m = 1$, then

$$P(1) := f_n = f_1 f_{n-1} + f_0 f_{n-2} \Leftrightarrow f_n = (1) f_{n-1} + (1) f_{n-2} = f_{n-1} + f_{n-2}$$

which is the definition of the Fibonacci numbers.

Inductive Step Suppose for some m that $P(m)$ is true, then we want to show that this implies that

$$P(m+1) := f_n = f_{m+1}f_{n-m-1} + f_m f_{n-m-2}$$

Beginning from the RHS of $P(m+1)$, we show that it is equal to the RHS of $P(m)$

$$\begin{aligned} f_{m+1}f_{n-m-1} + f_m f_{n-m-2} &= (f_m + f_{m-1})f_{n-m-1} + f_m f_{n-m-2} \\ &= f_m(f_{n-m-1} + f_{n-m-2}) + f_{m-1}f_{n-m-1} \\ &= f_m f_{n-m} + f_{m-1}f_{n-m-1} \end{aligned}$$

as desired. Thus by mathematical induction, $P(m)$ is true for all $n > m \geq 1$. $\square$

Remarks on m Notice that the m parameter can be chosen for a given n. This means we can represent a Fibonacci number as the sum of two products with factors all being smaller Fibonacci numbers, the values of which depend on m. Knowing this arithmetic meaning of m, we will use the m parameter from this proposition in some of the graph constructions that follow to explore its graphical meaning as well.

Fibonacci Cycles

The Fibonacci cycle defined below is constructed with an edge multiset to record the order and frequency of the edges that link the components of the cycle. After defining the cycle and stating its relationship with Fibonacci numbers, we'll show that these edges correspond to corresponding word.

Definition (Fibonacci Cycle): A Fibonacci cycle, $C_{f_n,M}$ is an undirected cycle graph with f_n vertices and edges constructed recursively as follows:

1. Let $M = \emptyset$ be a multiset. We create a recursive sequence s of K_2s and P_3s as follows:

   Set $s_0 = K_2$ and $s_1 = P_3$, where for both s_0 and s_1 we let one of their two vertices with degree 1 be called "left" and the other "right". Now for the recursive step:

   $$s_i = (s_{i-2}, m_i, s_{i-1}), i \in [2, n]$$

   where m_i denotes an edge incident to the right vertex of s_{i-2} with degree 1 and the left vertex of s_{i-1} with degree 1. We include m_i into M.

2. Define m_1 as the edge incident to both the left and right vertices of s_n. Include m_1 into M.

3. The resulting graph is a cycle, denoted $C_{f_n,M}$

Fibonacci Cycle Properties When $n \geq 2$,

1. $f_{n+2} = |V(C_{f_n,M})| = |E(C_{f_n,M})|$

2. $f_{n+1} = |E(C_{f_n,M}) - M|$

3. $f_n = |M|$

4. $f_{n-1} =$ number of P_3s in $C_{f_n,M} - M$

5. f_{n-2} = number of K_2s in $C_{f_n,M} - M$

6. There are f_{n-1} non m_2 edges in M

7. There are f_{n-2} m_2 edges in M. More generally, there are f_{n-i} m_i edges in M, $i \in [2, n]$

Fibonacci cycles for a given n therefore contain all Fibonacci numbers f_i where $i \in [0, n+2]$ in their structure.

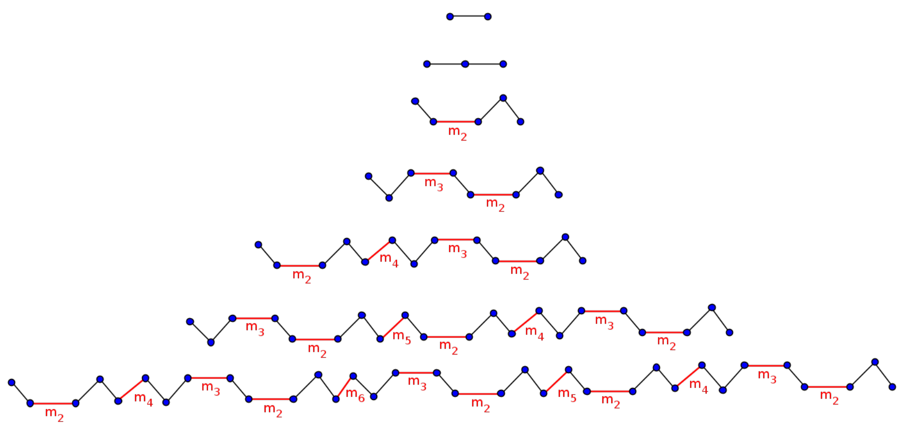


Figure 1: Each component from top to bottom are $C_{f_n,M}$s with the m_1 edge removed for $n \in [0, 6]$.

The Linking Edges Characterize an Interesting Class of Words Let w_n be the sequence of indices from the M edges of $C_{f_n,M}$ in cyclic ordering such that the index of m_1, 1, is the final entry. Then w_n has exactly f_{n-i} i elements for each $i \in [2, n]$ and exactly f_1 1.

Examples of W_n for $n \in [1, 6]$:

$$
\begin{aligned}
n = 1 &\Rightarrow w_1 = 1 \\
n = 2 &\Rightarrow w_2 = 21 \\
n = 3 &\Rightarrow w_3 = 321 \\
n = 4 &\Rightarrow w_4 = 24321 \\
n = 5 &\Rightarrow w_5 = 32524321 \\
n = 6 &\Rightarrow w_6 = 2432632524321
\end{aligned}
$$

We can construct these words in terms of word concatenation and morphisms as follows:

$$
w_n = w_{n-2}w_{n-1}, \text{with the morphism } 1 \rightarrow n, \text{ if 1 is not the final entry of } w_n. \tag{2}
$$

Characterizing the non-linker Edges of a Fibonacci Cycle Since the M edges of $C_{f_n,M}$ are always incident to either a K_2 or a P_3, between these linking edges of M are either a pair of adjacent edges in P_3 or a single edge of K_2. We can define another word w'_n that describes the cyclic ordering of these P_3s and K_2s where the edges in P_3 are denoted by "b" and the edge in K_2 can be denoted by "a". w'_n can be defined by the following morphisms applying to the elements in w'_{n-1}: $a \rightarrow b$ and $b \rightarrow ab$. These w'_n words are known in the literature as **Fibonacci words**.

Distribution of Letters in Fibonacci Words Ellis and Ruskey et al. in [2] proved that both the index sets of a and b, respectively in any w'_n are maximally even sets.[1]. A **maximally even** k-subset of $[t]$ is a set that is a translation of

$$\{\frac{ti}{k} : i \in [0, k-1]\} \tag{3}$$

Examples of Fibonacci words for $n \in [1, 6]$:

$$\begin{aligned} n = 1 &\Rightarrow w'_1 = b \\ n = 2 &\Rightarrow w'_2 = ab \\ n = 3 &\Rightarrow w'_3 = bab \\ n = 4 &\Rightarrow w'_4 = abbab \\ n = 5 &\Rightarrow w'_5 = bababbab \\ n = 6 &\Rightarrow w'_6 = abbabbababbab \end{aligned}$$

We can then characterize Fibonacci cycles in terms of w_n and w'_n by defining a new word W_n as follows:

$$W_n = (w'_{n_i} w_{n_i} : i \in [1, n])$$

Examples of W_n for $n \in [1, 6]$:

$$\begin{aligned} n = 1 &\Rightarrow w'_1 = b1 \\ n = 2 &\Rightarrow w'_2 = a2b1 \\ n = 3 &\Rightarrow w'_3 = b3a2b1 \\ n = 4 &\Rightarrow w'_4 = a2b4b3a2b1 \\ n = 5 &\Rightarrow w'_5 = b3a2b5a2b4b3a2b1 \\ n = 6 &\Rightarrow w'_6 = a2b4b3a2b6b3a2b5a2b4b3a2b1 \end{aligned}$$

Since W_n is an equivalent characterization of $C_{f_n,M}$, W_n has analogous relationships with the Fibonacci numbers:

Let $c(x)$ denote the frequency element x occurs in W_n. When $n \geq 2$,

1. $f_{n+2} = |W_n| + c(b) = 2c(b) + c(a) + |w_n|$
2. $f_{n+1} = c(b) + |w_n| = |W_n| - c(a)$, ie number of non-$a$ elements in W_n
3. $f_n = |w_n| = |w'_n| = |W_n|/2$
4. $f_{n-1} = c(b)$
5. $f_{n-2} = c(a)$
6. $f_{n-i} = c(i), i \in [2, n]$
7. $f_0 = c(1)$

[1] More precisely, [2] shows that Fibonacci words are a subclass of an object class they call Euclidean strings, but at the end of their paper they show that the index sets for the letters in Euclidean strings are maximally even sets.

Fibonacci Branch

Definition (Fibonacci Branch): B_{f_n} is a Fibonacci branch where

$$B_{f_1} = K_1, \text{which is called the } \textbf{root vertex};$$
$$B_{f_2} = B_{f_1}v, \text{where } v \text{ is adjacent to } B_{f_1}, \text{and is a } \textbf{non-root vertex}.$$

B_{f_n} is constructed from $B_{f_{n-1}}$ by adding to each non-root vertex with degree less than 3 a new adjacent non-root vertex.

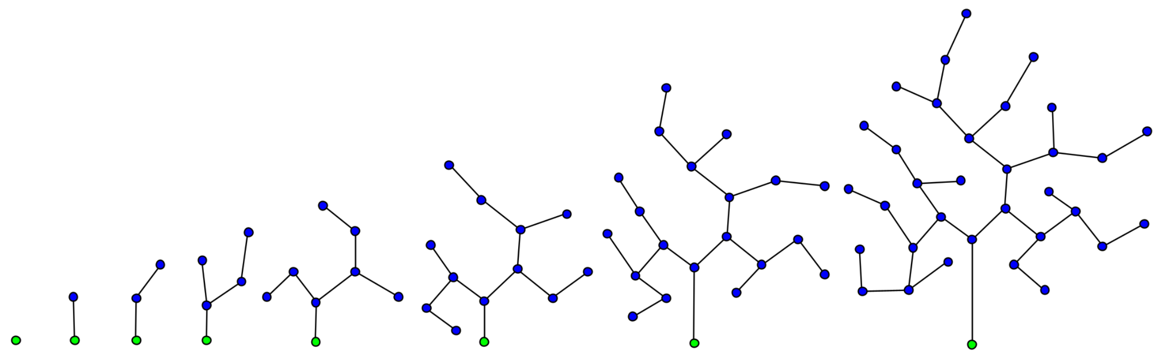


Figure 2: From left to right, B_{f_n} Fibonacci branches for $n \in [1, 8]$. Green vertices are roots and blue vertices are non-roots.

Fibonacci Branch Properties

1. When $n > 2$, B_{f_n} has:
   (a) f_n vertices and $f_n - 1$ edges.
   (b) $f_{n-2} - 1$ vertices of degree 3,
   (c) f_{n-3} vertices of degree 2, and
   (d) $f_{n-2} + 1$ vertices of degree 1.
2. $B_{f_{n+1}}$ has f_{n-1} more vertices than B_{f_n}

Branch Growth Let B denote the set of all Fibonacci tree branches.

Then define $g : B \to B$ algebraically as $g(B_{f_n}) = B_{f_{n+1}}$, and graphically by adding to each non-root vertex with degree less than 3 a new adjacent vertex in B_{f_n}.

Additionally, define $g^{-1}(B_{f_n}) = B_{f_{n-1}}$ graphically by removing all non-root degree 1 vertices from B_{f_n}.

g Invertibility $g^{-1}(g(B_{f_n})) = B_{f_n}$.

Since g only adds new vertices to existing vertices in B_{f_n} with degree 1, all degree 1 vertices in B_{f_n} are degree 2 vertices in $g(B_{f_n})$ and all newly added vertices in $g(B_{f_n})$ are degree 1, thus removing all these newly added degree 1 vertices, returns $g(B_{f_n})$ back to B_{f_n}. $\square$

Fibonacci Tree

Fibonacci Tree (Definition): Let $\tau = P_{f_m}$ be a path of size f_m, where $m < n$, then a Fibonacci tree $T_{n,m}$ is a tree with a central path τ, where $T_{n,m}$ is defined below:

First, define

$$S = B_{f_{n-m}}$$
$$L = B_{f_{n-(m-1)}}$$

where S is called a **small branch**, and L is called a **large branch**.

Second, define the initial cases for $T_{n,m}$

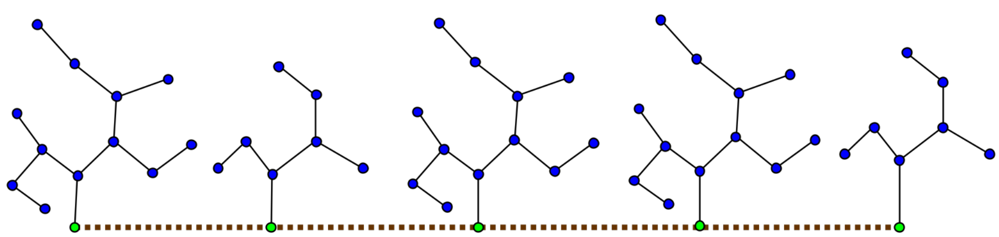
$$T_{n,1} = L$$
$$T_{n,2} = LeS$$

where $e \in E(\tau)$ is incident to the roots of both L and S. Finally, here is the recursive step:

$$T_{n,m} = T_{n,m-1} e T_{n,m-2} \tag{4}$$

Again, e is some edge in $E(\tau)$. The rightmost Fibonacci branch root in $T_{n,m-1}$ is adjacent to the left most Fibonacci branch root of $T_{n,m-2}$ through e.

Let τ be called the **trunk** of $T_{n,m}$.



Fibonacci Tree Properties These properties all follow from the proposition at the beginning of this document:

1. $f_n = |V(T_{n,m})| = |E(T_{n,m})| + 1$
2. f_m is the trunk size of $T_{n,m}$ (by definition)
3. f_{m-1} is the number of L Fibonacci branches in $T_{n,m}$
4. f_{m-2} is the number of S Fibonacci branches in $T_{n,m}$
5. $f_{n-(m-1)} = |L|$
6. $f_{n-m} = |S|$

Remarks on m In the context of Fibonacci trees, when m is larger (longer trunk), the branching in the Fibonacci tree is less complex because there are less vertices available to distribute across the branches, but when m is smaller, then the branching becomes more complex. m can therefore be seen as a measure of the level of branching in the tree inversely proportional to the size of its trunk, given n.

Fibonacci Tree Growth Let T be the set of all Fibonacci trees. Then define $g_t : T \to T$ algebraically as $g_t(T_{n,m}) = T_{n+1,m}$ and graphically as follows: apply the branch growth operation g_b to each branch $B \in T_{n,m}$.

Similarly, $g_t^{-1}(T_{n,m}) = T_{n-1,m}$ is defined by applying g_b^{-1} to every branch in $T_{n,m}$, but it is only defined when $m < n-1$, since $T_{n-1,m}$ does not exist when $n-1 = m$. The invertibility of g_t follows from the invertibility of g_b, so $g_t^{-1}(g_t(T_{n,m})) = T_{n,m}$, when $m < n-1$.

Super Fibonacci Tree

Definition (Super Fibonacci Tree): A super Fibonacci tree is like a Fibonacci tree except its branches are smaller Fibonacci trees with possibly their own trunks rather than just branches. We again take advantage of the proposition by using m to adjust branching complexity, but in super Fibonacci trees we'll be partitioning m into a vector M to get handle on the distribution of this branching complexity throughout.

Let $M = (m_1, m_2, ..., m_{f_m})$ be the sequence of positive integers denoting the trunk sizes of each Fibonacci tree incident to a trunk vertex of the super Fibonacci tree. A super Fibonacci tree is defined as a sequence:

$$T'_{n,\mathbf{M}} = (T_{j,m_i} : \begin{cases} j = n - m, & \text{if } w'_{m_i} = a \\ j = n - (m-1), & \text{if } w'_{m_i} = b \end{cases}, i \in [1, f_m]) \tag{5}$$

This sequence definition means that if Fibonacci trees are neighbours in the sequence, then their leftmost vertex in their respective trunks are incident to neighbouring vertices in $\tau(T'_{n,\mathbf{M}})$. The vertices in $\tau(T'_{n,\mathbf{M}})$ are the roots of its sub Fibonacci trees.

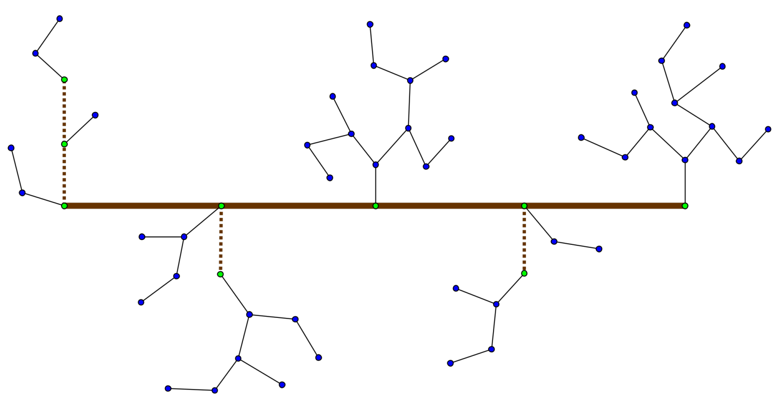


Figure 3: This is the $T'_{9,(3,2,1,2,1)} = (T_{5,3}, T_{6,2}, T_{6,1}, T_{5,2}, T_{6,1})$ super Fibonacci tree ($m = 4$). The solid brown edges form the super tree trunk, while the broken brown edges form the trunks of the sub Fibonacci trees. Notice that the third and fifth trees don't have their own trunks because $m_3 = m_5 = 1$, so these trees are actually equivalent to the Fibonacci branch B_{f_6}. Green vertices are roots while blue are non-roots.

Super Fibonacci Tree Properties

1. $f_n = |V(T'_{n,\mathbf{M}})| = |E(T'_{n,\mathbf{M}})| + 1$
2. f_{m-1} is the number of larger sub-Fibonacci trees of $T'_{n,\mathbf{M}}$ with size $f_{n-(m-1)}$
3. f_{m-2} is the number of smaller sub-Fibonacci trees of $T'_{n,\mathbf{M}}$ with size $f_{n-(m-2)}$

Remarks on M M provides a way of selecting a distribution of the branching complexity throughout a super Fibonacci tree.

Concluding Remarks

In this paper, we began with a well known proposition on Fibonacci numbers and explored some of its implications in cycle and tree constructions. The link between the Fibonacci cycle construction and maximal evenness was noted. Building off of the known construction of Fibonacci branches (Fibonacci trees in the literature), we generalized this construction twice over, because it was fun and m had interesting graphical meaning. Nice graph pictures were also viewed in the reading and writing of this document. I hope it was an enjoyable read! This was certainly fun to put together.

References

[1] N.N. Vorob'ev. Fibonacci numbers. Dover edition, 2011 (originally published 1961) (In particular, page 10)

[2] J. Ellis, F. Ruskey, J. Sawada, J. Simpson (2003). Euclidean Strings. Theoretical Computer Science 301, pp. 321-340.