Integer - Tree Isomorphism

Draft A

William Edwin Sharkey
WilliamSharkey@Gmail.com

November 14, 2013

Abstract

This document presents a simple method to map every positive integer to a unique isomorphic tree and the inverse - to map every tree to a unique integer. Mapping trees to integers allows for tree isomorphism checking. Trees assist in comparing and categorizing integers by graph properties. Tree representation of integers may find uses in teaching, communicating, or storing integers in computing systems.

I do not present a formal proof of a isomorphism, only an algorithm that suggests such.

1 Introduction

I became interested in mapping trees to integers when developing a bar-code system. The particular system uses a topological tree graph instead of familiar striped bar codes, and required a way to map trees to integers with a one-to-one correspondence. A critical property of this system is that orientation of the trees was to be ignored - only the isomorphic topology was to be significant. Not being able to locate an existing algorithm to assign trees to integers, I devised a method which is satisfactory for my purposes.

I found that integers have the properties of isomorphic trees when decomposed by prime factorizations, the communitivity of multiplication, and the inverted mapping between n and Prime(n).

2 Algorithms

2.1 Integer to Tree Algorithm

An algorithm for representing a positive integer Q as a tree is shown below, using an example integer, 165. This algorithm constructs the tree from the top-down, starting with a single node, Q.

1. Place Q at root node of tree. (EG: Q = 165)
2. Calculate the prime factorization of Q. (EG. 165 = 11 * 5 * 3)
3. Express each factor as P_i, where i is the index of the prime in the prime series. (EG: 165 = P_5 * P_3 * P_1 , it helps to have a table of primes at hand.)
4. For each prime factor, P_i, create a new child node, labeled i. (EG: Make three child nodes labeled 5, 3 and 1.)
5. Recur: for each child which is not equal to 1, recur. Repeat until every leaf of the tree is labeled with 1. When complete, you may want to re-draw the tree structure cleanly, and make obvious the root node location. Once the full tree is drawn, labeling nodes by values is redundant.

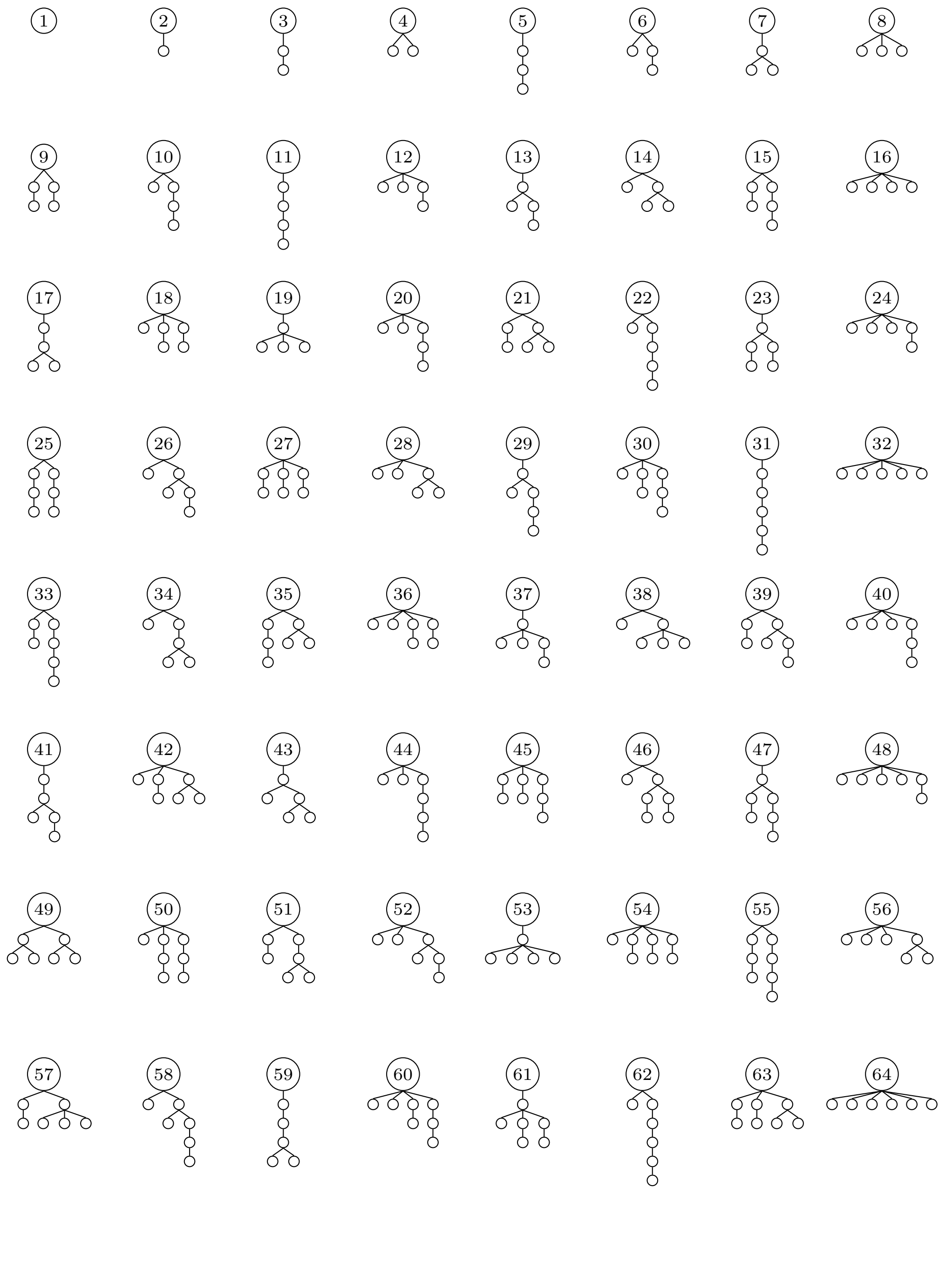
2.2 Tree to Integer Algorithm

An algorithm for representing a tree as an integer is shown below, using an example tree T. This algorithm constructs the tree from the bottom-up, starting with the leaves.

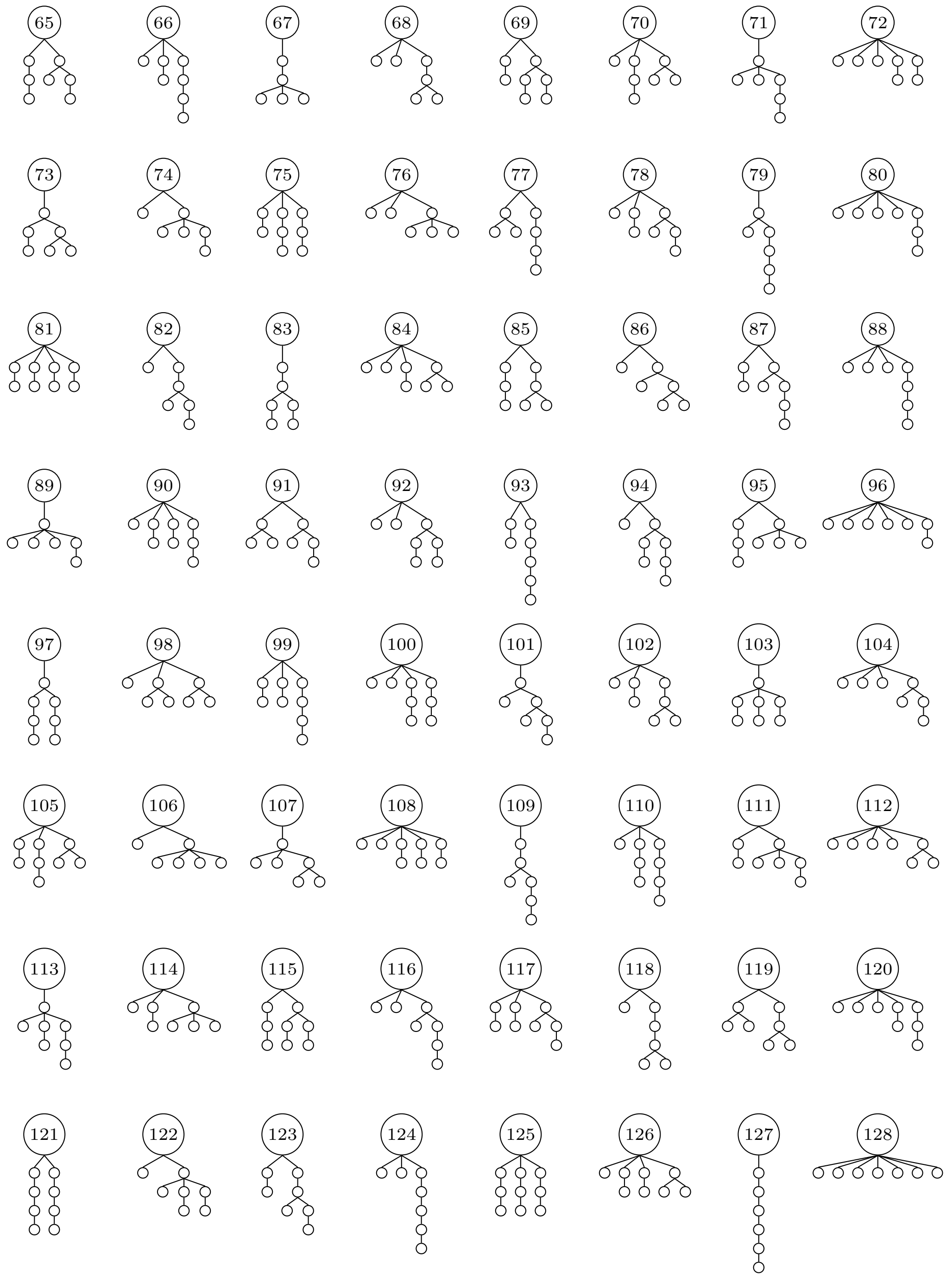
1. Do this only once: Find all leaves. Label them with the number one.
2. Find all vertices, v_n, which are unlabeled, but have all of their children, c_i, labeled.
3. Assign **v** the product of its children's corresponding i'th Prime. (EG: if **v** has children labeled 1, 2 and 5, then assign **v** = P_1 * P_2 * P_5 = 2 * 3 * 11 = 66.
4. Recur on step 2 until every node is labeled. When root is labeled, the process is complete. The label of the root is the integer associated with that tree.

3 Examples

3.1 Integer-Trees, 1 to 64

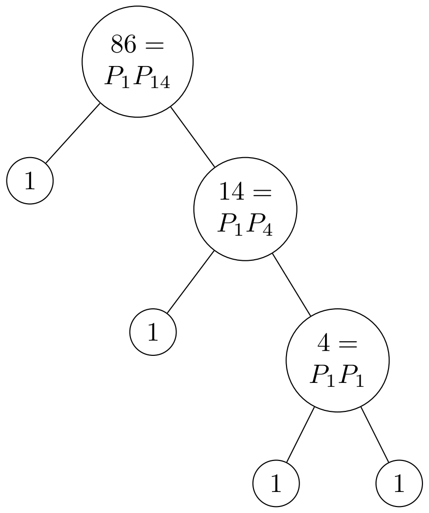


3.2 Integer-Trees, 65 to 128



3.3 Example: 86 Worked

Below is an example of the work a person may do to decompose the integer 86 into a tree.



3.4 Reference: Table of Small Primes

Index i	Prime(i)
1	2
2	3
3	5
4	7
5	11
6	13
7	17
8	19
9	23
10	29
11	31
12	37
13	41

Table 1: Prime numbers by index.

4 Comments

Please send comments, corrections and criticism to WilliamSharkey@gmail.com. Thanks for taking a look!