devised, and both subsequent ratio sums reduce by more than a factor of 2. Then if CNH is performed for the remaining division, it is very likely that this method will require fewer cuts than standard CNH. This is because the standard algorithm requires approximately ceiling(log2(a+b)) cuts, and factoring by an integer greater than 2 will reduce this logarithm. The following cut-trees demonstrate how this procedure can lead to fewer cuts than CNH. The desired ratio is 25:12, and the non-standard cut is 7:30.

The diagram makes it clear; if the first cut of 7:30 is made, then no matter which player takes the 7/37, the resultant ratio sum reduces substantially. And now the obvious question, how is it known that 7:30 is the optimal cut to make? One could exhaustively check each integer from 1 to (b–1) and determine which of these portions would yield the best reduction. This brute-force approach is certainly inefficient, especially if none of the options yields any reduction at all. Is there some way of determining whether or not a reduction is achievable, and if so, is there a technique for finding the exact cut that will do the job? These are all questions that Robertson and Webb ask in their text [1]. It turns out there is such a method to explicitly solve for more efficient cuts.

B. A Useful Result

Recall that a portion of the cake x is sought, with x < b, such that \((a–x)/b = n\), for some integer n, and \(a/(b–x) = m\), for some integer m. For a given ratio of \(a:b\), such portions x are given by the formula:

\[
x = a+b \mod (m*n), \text{ where } m|a \text{ and } n|b
\]

The term x will be named a “simplifier” for \(a:b\), as it simplifies ratio sums. The above result can then be derived. First, the properties of a simplifier are translated into the language of number theory. The result is a system of linear congruences, and the system can be solved accordingly. The derivation is shown below.

The diagram makes it clear; if the first cut of 7:30 is made, then no matter which player takes the 7/37, the resultant ratio sum reduces substantially. And now the obvious question, how is it known that 7:30 is the optimal cut to make? One could exhaustively check each integer from 1 to (b–1) and determine which of these portions would yield the best reduction. This brute-force approach is certainly inefficient, especially if none of the options yields any reduction at all. Is there some way of determining whether or not a reduction is achievable, and if so, is there a technique for finding the exact cut that will do the job? These are all questions that Robertson and Webb ask in their text [1]. It turns out there is such a method to explicitly solve for more efficient cuts.

B. A Useful Result

Recall that a portion of the cake x is sought, with x < b, such that \((a–x)/b = n\), for some integer n, and \(a/(b–x) = m\), for some integer m. For a given ratio of \(a:b\), such portions x are given by the formula:

\[
x = a+b \mod (m*n), \text{ where } m|a \text{ and } n|b
\]

The term x will be named a “simplifier” for \(a:b\), as it simplifies ratio sums. The above result can then be derived. First, the properties of a simplifier are translated into the language of number theory. The result is a system of linear congruences, and the system can be solved accordingly. The derivation is shown below.