Counting b-adic Fraction Sequences With a Sum of 1

Jiri Lebl

1 The Problem

We wish to count certain length sequences of b-adic fractions, that is fractions of the form $\frac{1}{b^n}$, that add up to 1, up to reordering. Let m be the length of the sequence, then for example when $m = 4$ and $b = 2$ we have the following sequences:

$$\frac{1}{2^2} + \frac{1}{2^2} + \frac{1}{2^2} + \frac{1}{2^2} = 1 \qquad \text{and} \qquad \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^3} = 1$$

We can really thus only concern ourselves with the exponents in those fractions, and so we could write two sequences as follows, always sorting from lowest to highest, as we do not care about the order:

$$(2,2,2,2) \qquad \text{and} \qquad (1,2,3,3)$$

We then wish to find a function $f(m,b)$ which would give us the number of sequences (up to reordering) of b-adic fractions of length m. From the above we can see that $f(4,2) = 2$. Furthermore we will define a function $s(n,b) = f\big((n-1)(b-1)+1, b\big)$ and we will see that this function describes all the cases of any interest and further, also has nice properties.

2 Preliminary Results

We first need some preliminary results about these sequencecs and a way to visualize them as trees.

Lemma 1 *If we take the highest exponent in any particular sequence, the number of occurences in the sequence of this exponent is divisible by b.*

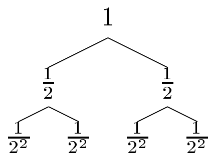
Proof: To do this we write down the sequence with exponents in increasing order:

$$\frac{1}{b^{e_1}} + \frac{1}{b^{e_2}} + \cdots + \frac{1}{b^{e_m}} = 1 \tag{1}$$

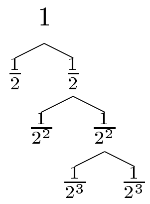
$$b^{e_m - e_1} + b^{e_m - e_2} + \cdots + 1 = b^{e_m} \tag{2}$$

From this we can see that both sides of equation (2) are divisible by b. The left side of (2) however contains a certain number of 1 terms. The number of these terms must therefore be divisible by b as well. The number of these terms is the number of occurances of the highest exponent in the sequence. ■

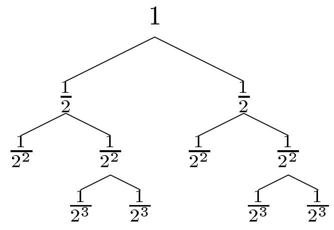
Now it will be useful to think of these sequences as full b-ary trees. First starting with the number 1 and placing 1 at the root of the tree, we can generate a sequence by successively taking a leaf of the tree and adding b branches to this leaf. We increase the exponent for each of these leaves. The terms of the sequence are then the leaves of the resulting tree. Thus the tree for $(2,2,2,2)$ and $(1,2,3,3)$ when $b = 2$ would look like:



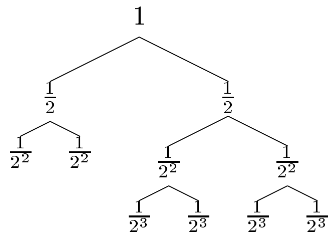
and



However we should note that just because two trees are not isomorphic to each other that the sequences are different. For example take these two trees:



and



They are non isomorphic yet they both give the same sequence $(2,2,3,3,3,3)$. So we can't just count the number of non-isomorphic trees.

Lemma 2 *All possible sequences can be represented by a tree.*

Proof: We will use induction on the highest exponent. If the highest exponent is 0, of course we can represent this sequences by a tree, since this is the trivial case. Similar if the highest exponent is 1 then the only possible sequence is $(1,1,\ldots,1)$, and this is also representable by a tree.

Now let's take a sequence with the highest exponent being $n+1$. By lemma 1 we see that the number of terms with exponents $n+1$ in the sequence is divisible by b. We could then produce a smaller sequence by just summing each b of these terms together to get a term of $\frac{1}{b^n}$. We do this until there are no more terms with exponents of $n+1$. We will thus get a sequence with the highest exponent is n which we know, by the induction hypothesis, to be representable by a tree. So if we reverse what we did by expanding each of the leaves that correspond to the new $\frac{1}{b^n}$ terms and give them each b branches labeling them with $\frac{1}{b^{n+1}}$, we will get a tree that represents this sequence. ■

Now notice that we cannot find any sequences for certain m and b, such that $m=2$ and $b=3$.

Lemma 3 *There are no sequences of length $m \neq n(b-1)+1$ for some n.*

Proof: Since all sequences are representable by trees, we can just take the tree corresponding to a sequence and start collapsing it's branches at vertices that only have leaves. Each time we collapse all the leaves of a branch, we loose b leaves and gain 1 (thus we lose b terms and gain 1) We can do this until there is only one leaf (the root) left. Since we can do this with any such tree, a tree will have to have $n(b-1)+1$ leaves, for some n, and thus sequences of other lengths do not exist, by invoking lemma 2. ■

Lemma 4 *Any exponent in the sequence is less then or equal to $\frac{m-1}{b-1}$*

Note that by lemma 3 the above fraction is always an integer.

Proof: To achieve the highest exponent we must build the deepest tree we can. This can be done by successively expanding only the deepest leafs of the tree. After this we will have exactly b exponents that are as high as possible and $b-1$ of all the other exponents (except for 0). This means that the highest exponent in this case is $\frac{m-1}{b-1}$, and this is then the upper limit for all sequences of lenght m as we constructed the deepest tree possible. ■

3 The h Function

We first define a helper h function before defining our real counting function.

Proposition 5 *The number of sequences of b-adic fractions of length m, that have the highest exponent of i and have at least $k \cdot b$ terms with the exponent i, for $m > 0$, $i \geq 0$, $m > 0$ and $b \geq 2$, is counted by the recursive function:*

$$h(m,i,k,b) = \begin{cases} m = b, i = k = 1 & 1 \\ i = 0 \text{ or } k \cdot b > m & 0 \\ \text{else} & \sum_{j=k}^{\lfloor m/b \rfloor} h(m - j(b-1), i-1, \lceil j/b \rceil, b) \end{cases} \tag{3}$$

Now the first case of h is where we are counting all the b-adic fraction sequences of length b where the highest exponent is 1. Obviously the answer is 1. Since $k > 0$ then when $i = 0$, there can be no sequences, since the only possible sequence with the highest exponent of 0 is just 1 itself, but then there cannot be at least $k \cdot b$ such terms. In case $k \cdot b > m$ there obviously can't be any sequences.

Now to handle the general case. Since we're counting the number of sequences that end in at least $k \cdot b$ terms with exponent i, we really want to count how many there are that end exactly in $k \cdot b$, plus how many there are that end exactly in $(k+1) \cdot b$, etc, until $\lfloor m/b \rfloor \cdot b$. To do this we just assume that there are exactly $j \cdot b$ terms with exponent i, where $k \leq j \leq \lfloor m/b \rfloor$. If we count all these cases separately we will have counted all the cases which we care about. So for any one of these j's, if we collapse all the branches with the leaves with exponents i, we will get a sequence of length $m - j(b-1)$ since we've collapsed $j \cdot b$ leaves with exponents i into j leaves with exponents $i-1$. This sequences will then have the highest exponent of $i-1$, and there are at least $\lceil j/b \rceil \cdot b$ terms with the exponent of $i-1$. These we can again count with the recursive use of the h function using $h(m - j(b-1), i-1, \lceil j/b \rceil, b)$. We can then just some up all these cases for all the j's to get our result.

4 The f Function

Proposition 6 *The number of sequences of b-adic fractions of length m is counted by the function $f(m,b)$, for $m \geq 1$ and $b \geq 2$, defined as:*

$$f(m,b) = \begin{cases} m = 1 & 1 \\ m > 1 & \sum_{i=\lceil \log_b m \rceil}^{\frac{m-1}{b-1}} h(m,i,1,b) \end{cases} \tag{4}$$

If the definition of h is correct then it is easy to see that then f would properly count the sequences as advertized, we just separately count all sequences with the same highest exponent i. The lowest exponent we need to worry about will be $\lceil \log_b m \rceil$ since that's the depth of the shallowest b-ary tree possible. The highest exponent is $\frac{m-1}{b-1}$ by lemma 4. So we just handle each possible exponent separately and sum the result. Exception is the case where $m = 1$ where the highest exponent is 0, but we handle that case separately.

5 The s Function

From lemma 3 we know that for when $m \neq k(b-1)+1$, then there is no such sequence for that combination of b and m, so f is zero at all those points. We could then define a function that avoids those points altogether. We therefore define a function $s(n,b)$ as follows:

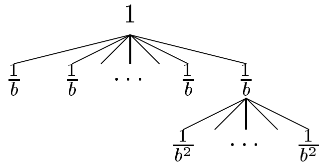
$$s(n,b) = f\big((n-1)(b-1)+1, b\big) \tag{5}$$

This way we get $s(n,2) = f(n,2)$. We can also notice one thing about s, each time we increase n, we have branched some leaf into b more leaves with a higher exponent. So the s function tracks how the trees are growing. For example at $n = 2$ we are for all b at a point where there are b leaves stemming from the root. At $n = 3$ there are $b - 1$ leaves stemming from the root, plus one internal vertex which itself has b leaves. Note thus that for $n = 2$ and $n = 3$ the s function is always 1 no matter what b is.

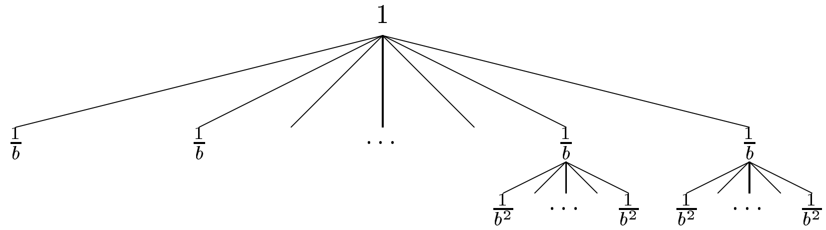
FIXME: more on s, pictures, tables

Theorem 7 *For $2 < n < b + 3$, $s(n,b) = 2^{n-3}$*

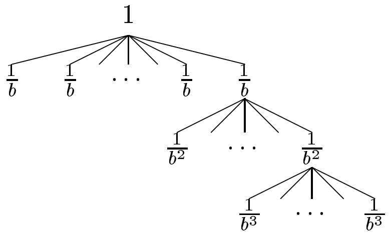
Proof: We first start at $n = 3$, obviously $s(3,b) = 1 = 2^{3-3}$. We now have two levels in the tree, one where there are $b - 1$ leaves with exponent 1 and one where there are b leaves with exponent 2.



We now define an algorithm to successively generate all the possible sequences. We always concern ourselves only with the deepest level in each tree. We can add one more fork to the current deepest level. The new tree will have the same highest exponent.



We can also add a level by adding b branches onto one of the deepest leaves. This would increase the highest exponent by 1.



Now as long as we have a place to go on the current deepest level, we will always generate 2 new distinct trees from every old tree. We can always add a new level, so the amount of free leaves at the current level is what limits us. The first place we will run out of free leaves is quite obviously the first level where we begin with $b-1$ free leaves. So we can only double the number of sequences $b-1$ times. Thus, $s(n,b) = 2^{n-3}$ only for $2 < n < b+3$. ■

Corollary 8 *For* $n = b+3$ *and* $n \geq 3$, $s(n,b) = 2^{n-3} - 1$

Proof: By the same idea as the previous proof, we will have a power of 2 growth until $n = b+3$. At this point the lowest level will run out of free leaves. But since all the other levels still have empty free leaves, we can still create 2 new trees everywhere else. So we fail in only one case, that is we can add a new level, but not expand the current level. And thus for $n = b+3$, $s(n,b) = 2^{n-3} - 1$. ■