The Math Less Traveled

Explorations in mathematical beauty

Challenge #12: sums of powers of two

Posted on April 18, 2008 by Brent

A few interesting problems for you to think about:

1. Given a positive integer *n*, in how many ways can *n* be written as a sum of powers of two, when each power is allowed to occur at most once? For example, 11 can be written as $11 = 2^3 + 2^1 + 2^0 = 8 + 2 + 1$. But we're not allowed to write $11 = 2^2 + 2^2 + 2^1 + 2^0$, since 2^2 occurs twice. The order of the terms in the sums don't matter, so $2^3 + 2^1 + 2^0$ and $2^1 + 2^3 + 2^0$ don't count as different sums.
2. Now what if we allow at most two copies of each power of two? Both of the examples for 11 shown above would be allowed now. Say $h(n)$ denotes the number of different ways to write *n* as a sum of powers of two with at most two copies of each power allowed. What can you say about the function *h*?
3. What if we allow an unlimited number of each power of two? Now how many ways are there to write *n* as such a sum?

Post your solutions in the comments (partial solutions are welcome and encouraged), but only if you didn't know the answer as soon as you read the question. =)

About Brent
Associate Professor of Computer Science at Hendrix College. Functional programmer, mathematician, teacher, pianist, follower of Jesus.
View all posts by Brent →

This entry was posted in challenges, counting, pattern and tagged counting, partitions, powers of two, sums. Bookmark the permalink.

18 Responses to *Challenge #12: sums of powers of two*

Jonathan *says:*

April 19, 2008 at 9:48 am

I haven't seen this, but it looks like fun. After a few minutes playing with both, I decided to set them aside, but share what I noticed.

Question 1 I do know, and won't comment.

For only 2 copies allowed, I notice, strangely enough, that it goes up and down again. For example, 4 is 4 or 22 or 211 (I am writing the value of the powers, and omitting the +), but 5 is only 41 or 221 (as 2111 and 11111 are not allowed). There are 3 ways to write 6, but only 1 to write 7. Even more dramatic.

For an unlimited number of copies allowed, the answer for an odd number matches the answer for the preceding even number.
4: 4, 22, 211, 1111
5: 41, 221, 2111, 11111
I've just added a 1 to each way to write 4.

For this question, the number of ways rises every time we reach a new even number.

Anand *says:*

April 20, 2008 at 3:15 pm

A response to Question 3:

Let's say S(n) is the number of ways to write n as the sum of non-negative powers of 2. Then you can calculate it recursively using the following:

S(1)=1, and S(2)=2
For odd numbers: S(2n+1) = S(2n)
For even numbers: S(2n) = S(2n-1) + S(n)

(i) Jonathan explains above why S(2n+1) = S(2n).

(ii) Now, to see why S(2n) = S(2n-1) + S(n):

Let's say you take any even number, 2n, and list out ALL the ways of writing it as the sum of powers of 2. Here's the list using the example of 2n=6:

2,4
1,1,4
2,2,2
1,1,2,2
1,1,1,1,2
1,1,1,1,1,1

Now, split that into two lists: sums that included at least one "1", and sums that did not. The trick is now to find out long each of these lists is, and then add them up.

First, let's look at the list where each sum has at least one "1" in it. For each sum in this list, just remove one of the "1"s, and you're left with exactly the list of ways to sum up to 2n-1. (This is a nontrivial claim, but I didn't want to over-complicate the solution.) So this list is S(2n-1) long.

Next, look at the list where each sum has no "1"s in it. For each sum in this list, just divide every term by 2, and you're left with exactly the list of ways to sum up to n. So this list is S(n) long.

Finally, then, S(2n)=S(2n-1)+S(n).

Check it on the example: S(6) = S(5) + S(3) = S(4) + S(2) = ...

What I'm now wondering is: what is the asymptotic growth of S(n)? I.e., is there a simple function f such that S(n) is "big oh" of f?

Jason Dyer *says:*
April 20, 2008 at 7:52 pm

For question #2, I have a way of expressing all the ways of counting as a tree.

Turn the number n into a binary form. Let k be the number of digits. You are going to have a tree with k layers; label each layer of the tree with the digits from n. For example, 109=1101101 so the layers would be 1, 0, 1, 1, 0, 1, 1.

What we're imagining is adding two binary numbers together.

We will be working from right to left, imagining we are in a state of "carry" (carrying 1) or "not carry" (no carry of 1).

Start at the top of the tree with a node labelled "not carry".

When the layer currently at is a 1, if the state is "not carry" the node will go to the next level without a split, and the state of the next node is "not carry". (1 + 0 = 1)

If the layer is a 1 and the state is "carry", the nodes split at the next level to one "carry" and one "not carry". (1 (carry) + 1 + 1 = 11 or 1 (carry) + 0 + 0 = 01)

If the layer is a 0 and the state is "not carry", the nodes split at the next level to one "carry" and one "not carry". (1 + 1 = 10 or 0 + 0 = 0)

If the layer is a 0 and the state is "carry", the node goes down to the next level without a split and the state is still "carry". (1 (carry) + 0 + 1 = 10)

The last layer will be a 1 (since we wrote the number in binary and we don't need any leading zeros) and there is no bifurcation.

Count the number of leaves in the tree, and we have the number of ways of forming the sum.

Steve Gilberg *says:*
April 21, 2008 at 11:14 am

This is reminding me of that fourth-grade book "Math for Smarty Pants." In one of the included cartoons, two kids are finding ways to express numbers (read: positive integers) as sums of consecutive others. I quickly grew bored of that activity, but I was interested to figure out rules for it. Eventually, I figured out that the only positive integers that could not be expressed as sums of consecutive positive integers were powers of 2.

Jason Dyer *says:*
April 21, 2008 at 11:25 am

As a quick addendum, if you don't care about the tree structure but want the total possible sums, start with two variables C=0 and NC=1. Start counting the binary version of the number n from right to left. If you hit a 1, add C to NC. If you hit a 0, add NC to C. By the end, C + NC = the number of possible sums.

I'm not sure if it is possible to express this explicitly rather than algorithmically because the actual order of the digits matters. For example, there aren't any bifurcations at all until the first 0 hits. (So it keeps adding C (0) to NC (1) so NC = 1 + 0.)

(This is all presuming I didn't make some fatal error in my above procedure, of course.)

Brent *says:*
April 21, 2008 at 2:49 pm

Jonathan: the fact that #2 isn't monotonic (goes up and down) is curious indeed! It's not intuitively clear to me why this should be so, but… there it is. =)

Anand: Great analysis of #3! Your question on the asymptotic growth of the sequence is interesting, since according to this, the precise asymptotic growth is not actually known! An analysis similar to yours is also possible for #2.

Jason: Wow, that's a really cool and creative analysis there! It seems to work. Now to figure out why… =)

Nick Johnson *says:*
April 25, 2008 at 10:26 am

Maybe it's just a coincidence, but I notice that h(n) contains palindromic sequences of increasing length, delimited by 1s:
1 1 2 1 3 2 3 1 4 3 5 2 5 3 4 1 5 4 7 3 8 5 7 2 7 5 8 3 7 4 5 1
Contains sequences of length 0, 1, 3, 7, 15 (eg, 2n+1).

Brent *says:*
April 25, 2008 at 12:36 pm

Nick: excellent observation! As it turns out (and as is often the case when one discovers an apparent pattern like this), it is definitely NOT a coincidence, and I'll post something in the near future that can shed some light on why this happens.

Ray *says:*
June 17, 2008 at 9:31 pm

For question 3, the answer is that every positive integer has infinitely many representations.

We demonstrate by showing a distinct representation of 1 for every integer n. (Extending this to other integers is left as an exercise to the reader.)

$1 = 2^0$
$1 = 2^{-1} + 2^{-1}$
$1 = 2^{-2} + 2^{-2} + 2^{-2} + 2^{-2}$

(and in general)

$1 = (2^n$ copies of $2^{-n})$

Further consideration should reveal that the answer to question 1 is, in fact, one more than you probably thought it was when you first read the question (since 0.9999... = 1).

🙂

Brent *says:*
June 18, 2008 at 12:29 am

Ray: aha, excellent! When I posed the problems, I was intending to restrict solutions to nonnegative powers of two... but I guess I didn't actually say that, did I? =)

Dave *says:*
March 28, 2009 at 4:12 pm

For #2:

$h(n) = 1$ for all $n = 2^x - 1$, where x is an integer >= 1. There are similar patterns for $h(n) = 2$, $h(n) = 3$, and so on... I am still working on the others.

I'm sure there is some connection between all the patterns, as well, but that is yet to be seen.

Dave *says:*
March 29, 2009 at 1:36 pm

For #2:

$h(n) = 1$ for all $n = 2^x - 1$, where x is an integer >= 1
$h(n) = 2$ for all $n = 3(2^{(x-1)}) - 1$ where x is an integer >= 1

...still working on the others.

Dave *says:*
March 29, 2009 at 3:44 pm

For #2:

h(n) = 1 for all n = 2^x – 1, where x is an integer >= 1

h(n) = 2 for all n = 3(2^(x-1)) – 1 where x is an integer >= 1

h(n) = 3 for all n = 5(2^(x -1)) – 1 and n = 7(2^(x-1)) – 1, where x in an integer >= 1

Dave *says:*

March 29, 2009 at 6:10 pm

For #2:

See above entries for h(n) for n = 1,2,3

h(n) = 4 for all n = 9(2^(x-1)) – 1 and n = 15(2^(x-1)) – 1

I believe I have the pattern:
Notice that, in each case (except for h(n) = 1), all that is changing are the coefficients. For h(n) = 2, the coefficient is 3.
For h(n) = 3, the coefficients are 5 and 7. For h(n) = 4, the coefficients are 9 and 15, and so on….

So, if we want to know the value of h(n), we can just break down (n + 1) to it's prime factorization. Whatever the greatest factor that is not a power of 2 is, that will be the coefficient (There has to be a better way to word that, so I apologize to the reader).

For example, if we want to know h(27):
(n + 1) = 28, and 28 factors to 7 * 2^2.
So we're looking for the function with a coefficient of 7, which tells us that h(27) = 3 (see above posting for appropriate function).

This example was easy because we already knew that h(n) = 3 corresponds to a coefficient of 7. But what if we arrive at a coefficient that we don't already know the function for. There must be some general way to find the coefficient for any h(n) = k.

After h(n) = 1, each h(n) has 2 corresponding coefficients. The formulas for finding these two coefficients are as follows:
h(n) = k has coefficients 2^(k-1) + 1 and 2^k – 1.

So, now we can find the coefficients for any h(n).
Examples: h(n) = 5 has coefficients 2^(5-1) + 1 = 17 and 2^5 – 1 = 32.
h(n) = 6 has coefficients 2^(6-1) + 1 = 33 and 2^6 – 1 = 63, and so on…

Now it should be simple to find h(n) for any n.

Example 1:
h(33) = ?
(n+1) = 34, and 34 factors to 17 * 2^1. We're looking for a coefficient of 17, which should yield a solution in one of the two "k" equations:
2^(k-1) + 1 = 17
2^(k-1) = 16
2^(k-1) = 2^4
k-1 = 4
k = 5, so h(33) = 5.

Example 2:
h(71) = ?
(n+1) = 72, and 72 factors to 9 * 2^3. We're looking for a coefficient of 9:
2^(k-1) +1 = 9
2^(k-1) = 8
k-1 = 3
k = 4, so h(71) = 4.

Example 3:
h(129) = ?

(n+1) = 130, which factors to 65 * 2^1. We're looking for a coefficient of 65:

$2^{(k-1)} + 1 = 65$

$2^{(k-1)} = 64$

$2^{(k-1)} = 2^6$

k-1 = 6

k=7, so h(129) = 7.

Example 4:

h(63) = ?

Special case, since 63 is of the form $2^x - 1$. So h(63) = 1.

Unfortunately, as you try this algorithm repeatedly, you'll notice that it breaks down for many numbers. If anyone can explain why, I would appreciate it.

Dave *says:*

March 29, 2009 at 9:13 pm

In my last post, I mentioned that my algorithm breaks down for many values of n. I carried out the method for values of n from 1 to 31, and it does not work for h(10), h(12), h(18), h(20), h(21), h(22), h(24), h(25), h(26), and h(28). Does anyone know why the algorithm breaks down for these values?

Dave *says:*

March 29, 2009 at 9:48 pm

I think I may have found the answer to my own question:

I mistakenly assumed that each k had only two coefficients. Now, I realize that h(n) = 5 has 4 coefficients: 11, 13, 17, 31. This takes care of my problems with h(10) = 5 and h(12) = 5.

h(n) = 7 and h(n) = 8 also have more than 2 coefficients, but I have not found the others yet….

I wonder if there is a way to find out how many coefficients exist for each h(n) = k?

Pingback: *Challenge #12 solution, part II « The Math Less Traveled*

Pingback: *More hyperbinary fun « The Math Less Traveled*

The Math Less Traveled

Blog at WordPress.com.