

YouTube Summarizer

Transcript:

Hey guys, Greg here. Let's solve base seven, Lico number 504. So pretty simply, we're given an integer called num and you need to return a string of its base seven representation. So we are given a number like 100. The base seven representation would be 202. We'll soon see this is going to be two times seven squared plus two, which is equal to 100. And as you can see, we can have negative numbers as well. So let's say we are given a number like 100. Now this is in base 10. So what that really means is that this is the ones column, this is the tens column, and this is the hundreds column. And so in exponents, this is actually the zero, the one and the two. It's going to be one times 10 to the two plus zero times 10 to the one plus zero times 10 to the zero, which as we know, of course, is just equal to 100. So this is a base 10 number. But to convert a two base seven, what you need to do is keep dividing it by seven. So if you take 100 and you divide that by seven, now just to do some quick mental math here, I know that seven times 11 is equal to 77. So that means that seven times 12 would be an additional seven, which is 84. That means that seven times 13 is going to equal 91. And if you do one more, so seven times 14 would be 98 here. If you added any more sevens, that's going to bring you outside to 105. And so that is too high. So what that tells us here is that 100 long division by seven is going to give you 14. And then the remainder of that is going to be 100 minus 98, which is equal to two. So you have a remainder of two. Now we're not done yet. We need to take this 14, divide that by seven. We're going to get an even two with a remainder of zero. We then take our two, we divide that by seven. We're going to get zero, remainder of two. And so we stop when this thing is at zero. Now this tells us that our base seven representation is basically going this way of the remainders. So it's going to be two then zero then two. So this is our base seven representation. Okay, now how do we confirm that this is actually true here? Well, what it means is that this is going to be again, the zero exponent. This will be the one and this will be the two. So what it means is that we're going to have two times seven squared plus zero times seven to the one plus two times seven to the zero. So this will give us two times 49 plus zero times seven plus two times one. This is going to give us 98 plus zero plus two, which is equal to 100. So we can see that that works. And by the way, when you write stuff with a base other than 10, generally you write it down here. So this is base seven and we don't generally write it, but this would be base 10. Now the runtime of this algorithm, basically what we're doing is just we're repeatedly dividing by seven. If you keep repeatedly dividing by a factor, that's exactly what a logarithm is. So this is going to run in big O of log base seven. We start with a number like N and we keep dividing that by seven. So this is very, very fast. And again, since we're getting one every time we divide by seven, it's basically going to be exact to the same thing here. We'll have roughly log base seven if N thing is to store. Okay, so just a quick base case here. If the number we're dealing with is zero, then we just want to return zero. And what we'll do is while the number is greater than zero, we're going to keep dividing it. So look at the remainder is going to be the number modulo by seven. And we want to keep track of our remainders. So I'm actually going to make an empty list out of this called remainders just an empty list. And then what we can do is just remainders.append with the string representation of the remainder. So this is just going to be a list of our remainders in order. And then you would just divide that number by seven here. So what this says here is num divide divide equals seven. It sets num equal to num divided by seven. And we use the double slash to make sure that it's integer division. Okay, now as we saw, we would actually need the opposite order of our remainders. So we'll just do remainders.reverse, which is going to reverse that list in place. Then you would just need to join this into a singular string. You would do return the empty string dot join of the remainders. And it's going to glue all of those remainders into a single string. Now this would work for positive numbers as well as zero. But as you'll see when we run this, this is actually going to give a wrong answer for minus seven. It says that we outputted

nothing. And the reason for that is because we're saying while the number is greater than zero, if the original number is less than zero, then this thing isn't really gonna run at all. So we're gonna change this up a little bit. What we're gonna do is keep track of the original number, which will just be a copy of the number. And then what I'm gonna do is set the number equal to the absolute value of the number. So now this while loop is going to run and accumulate that magnitude. However, we just need to say if the number was originally negative, put a negative on the beginning. And actually if we do it before we reverse it, that's gonna put it at the end. And so when we reverse it, it will make the negative at the beginning. So if the original num is less than zero, then we want to remainders dot a panned with the negative sign. So again, if our original number was negative, we simply just place a negative on the end here so that when we reverse it, it will be the negative version of that number. So now when we run this, this is going to pass that case. And if you submit it, it's gonna pass all of our test cases very quickly. I hope this was helpful guys. Check out [algomap.io](https://www.algomap.io) in the description. If you haven't already and have a great day folks, bye bye.

Summary

We're given an integer called num and you need to return a string of its base seven representation . To convert a two base seven number, what you need is to do is keep dividing it by seven . If the number we're dealing with is zero, then we just want to return zero . If you add any more sevens, that's going to bring you outside to 105. And so that is too high. If you added any more 7s that's too high, that will bring you out to 105 .