Jan 1, 2022

A CASE OF ADHOC PROBLEM SOLVING

BY KALPANA BAHETI

What is the largest number divisible by 11, comprised of digits 0 to 9, where each digit appears exactly once?

Note: Recall that a number is divisible by 11 if the difference in the two sums created by alternate digits is divisible by 11. For example – 12624381 has the alternating sums – 19 from 1+6+4+8 and 8 from 2+2+3+1, and the difference is 19-8 which 11, and that's divisible by 11, so 12624381 is divisible by 11.

Before we begin, let's establish certain observations we will use to solve this problem ~

1. From the note given, there is another way of summation and differences. **We take the differences of two consecutive digits each and add these differences and test whether THAT sum is divisible by 11.** It's still the same test.
Example – 12624381 has the differences – 1-2 = -1, 6-2 = 4, 4-3 = 1, 8-1 = 7. And we add those differences, we get: -1 + 4 + 1 + 7 = 11, which is divisible by 11.

2. Based on the perspective we've adopted in (1), for the digits 0 to 9 to be arranged, **we have 10 digits in total and hence 5 differences to add that should sum up to a multiple of 11**.

3. We want the largest number possible. That means taking any two consecutive numbers, we'd want the larger one to be before the smaller one (for example – 83 is greater than 38). **Hence, we'd want ALL of our five differences to be positive.**

4. Again, we want the largest number possible. We'd like our consecutive numbers to have the least difference possible, and since no digit is repeated, the least difference is 1. **Hence, we try to maximize the number of differences that are 1.**

5. For a sum of 5 numbers to add up to 11 (as described in observation 3), we can ensure this much –

a. *The sum must be even + odd, since 11 is odd.*

b. *There is only one way that odd part of (a) can be created and that is through adding an odd number of odd differences from those 5 differences.*

c. *Once you do (b), we have 5 – (some odd number) left which is even, since 5 is odd, and so we have an even number of even differences in those 5 differences.*

Hence, out of our five differences to be summed, there are an odd number of odd differences, and an even number of even differences.

If all our five differences were 1, we'd have – 1+1+1+1+1 = 5 and that's not divisible by 11. Now if we wanted it to be divisible by 11, we'd have to either bring it to multiples of 11 less than five – which would involve negative differences, or multiples greater than 5.

We don't want to use negative differences since that brings down our largest value. So, we tweak these five differences to meet the first positive multiple of 11 which is 11. According to the divisibility rule, we could check for greater multiples of 11 as well, but we needn't, and I'll explain why in the end.

Now, let's begin.

I have the initial sum 1+1+1+1+1, and to make it sum up to 11, **while retaining as many 1s, as possible (according to observation 4)**, I need to make it – 1+1+1+1+7.

This would be a good time to address a very pertinent question – how do we know such a 10-digit number even exists (with digits 0 to 9) that can create these differences through consecutive numbers?

That brings us to observation 6 –

6. Considering we have five differences adjusted over digits 0 to 9, and that follow observation 5-

1. *To fit a single difference that isn't equal to 1, and that's created from* ***a-b*** *where a and b belong to [0,9] – you ensure that your higher value, which is a, leaves an even number of spaces from the top – as in,* ***9-a*** *should be even. Then from all such (a,b) possibilities, you pick the one where a is the minimum because that leaves greater number of consecutive digits greater than a, that can enlarge your number.*

2. *To fit multiple differences that aren't 1, but are all the same number, start your first difference with the same rule as 1, and keep the upper limit minimum. Then stack the other difference pairs upon each other. For example, if your first difference is created from a – b, and you have two other differences like that – your other differences are made from (a+1) – (b+1) and (a+2) – (b+2), and likewise for three differences and so on.*

Consequently, you will be able to adjust your difference values that are not equal to 1, by following these rules, and STILL ensuring they sum up to 11, AND MAKE SPACE FOR THE DIFFERENCES THAT ARE EQUAL TO 1.

Now that we've established the conditions under which we are guaranteed to generate multiples of 11 from digits 0 to 9 while still keeping 1s as differences, let's get back to our problem -

1+1+1+1+7. For the difference 7, we have the following arrangements possible from digits 0 to 9 which are (7-0) or (8-1) or (9-2). According to observation 6, (8-1) is out, since it does not leave an even number of digits after 8, ie, 9-8 = 1. So, we have (7-0) and (9-2). And we pick (7-0) because it has the lowest upper limit, leaving space for one consecutive pair of digits greater than 7 which is 98. Hence, we get the multiple – 9870654321, where the differences of 1 came from 98, 65, 43, and 21. You see now why observation 6 was made to ensure even space from top?

We also see that the number of differences equal to 1, were 4, but we couldn't keep consecutive numbers in this manner – 98 76 54 32, since the difference that was NOT equal to 1, was so high. It was 7.

So, we trade-off through maximizing the number of 1s in our differences and reducing our maximum difference that isn't 1.

Let us establish one last observation before we solve our problem –

7. While abiding with observation 5, if I had to build k differences that are NOT equal to 1, such that these k differences add up to n, then to minimize the maximum value from these k differences I should use the differences n/k.

In the case where n/k isn't a whole number, I take the floor and ceiling values alternatively for n/k and use those as differences.

Example – If n = 8, and I had 4 slots to fill, I'd assign each of four differences to 2,2,2,2. Because if I try to reduce one of them, another will shoot up, and that will affect my overall arrangement of digits from 0 to 9, as we saw with the case of – 1+1+1+1+7.

This way I can minimize the maximum value of k differences that add up to n, which is what we need for the trade-off.

And now we have exactly 4 combinations to test for the tradeoff –

1. 1+1+1+1+7, which yields – **9870654321** (according to 6.1)

2. 1+1+1+4+4 (according to observation 7 and 5), which yields – **9876514032** (according to the stacking rule of observation 6.2).

3. 1+1+3+3+3 (according to observation 7 and 5), which yields – **9876524130** (according to the stacking rule of observation 6.2).

4. 1+2+3+2+3 (according to observation 7 and 5), and we don't have a rule for this in observation 6, but we don't need to worry since with the number of consecutive pairs from the top, cases 1, 2 and 3 have already beaten it.

SO! We have reduced our case of 10P10 combinations of digits 0-9 to test (maybe 10P10 x 0.9, if you removed all combinations starting with 0), to **four cases** by deduction.

We now see from these four cases that case 3 – **9876524130 is the highest possible number satisfying our problem statement.**

And we've established WHY without any trial and error, or combinatorics used, only basic number theory principles.

NOTE 1 -

A final point to make is that we didn't try for other multiples of 11 once we knew it was possible to adjust to 11 by proof, because to maximize the differences equal to 1 on other multiples of 11, would've meant that the remaining differences that weren't 1 would either have not been possible with our digits, or would've shot up and taken away even the least we got from summing up to 11 – the least being a multiple that started with 98.

NOTE 2 –

This is a more important thing to observe. We could've saved a lot of mental energy on our part by brute force combinatorics or even semi-deduction. However, this solution used NO combinatorics. If a system were to solve this, it should be taught how to apply these number theory deductions which would make it A LOT more efficient and accurate on such problems, even though it would be challenging to initially design such a system. But remember that this was 10 digits with a specific clause. If it were 100 digits with other criteria, a system cannot run that many combinations, and must learn to use such proof-driven deductions to reduce the trial pool.