

Arbitrary Precision Increment

In this lesson, you will learn how to solve the problem of the Arbitrary Precision Increment using arrays.

We'll cover the following



- Algorithm
- Implementation
- Explanation

In this lesson, we will be considering the following:

Given: An array of non-negative digits that represent a decimal integer.

Problem: Add one to the integer. Assume the solution still works even if implemented in a language with finite-precision arithmetic.

We will cover how to solve this problem algorithmically, and then code a solution to this question in Python.

Arbitrary Precision Increment

Array A

1	4	9
---	---	---



Adding 1



Array A

1	5	0
---	---	---

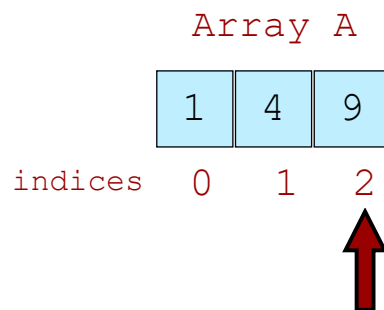
Algorithm

For an array **A**,

1. Add 1 to the rightmost digit.
2. Propagate carry throughout the array.

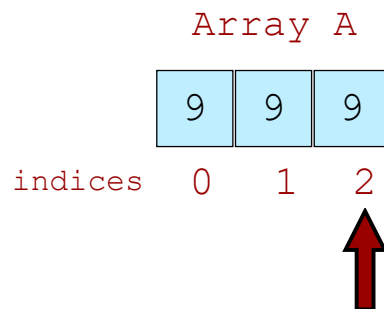
This will be similar to the “standard grade school” approach.

Arbitrary Precision Increment : Example 1



Add 1 to the rightmost digit

Arbitrary Precision Increment : Example 2



Add 1 to the rightmost digit

1 of 6



Implementation

In Python, you can solve the problem with just one line.

```
A = [1, 4, 9]
s = ''.join(map(str, A))
print(int(s) + 1)
```



Explanation

On **line 2**, the `map` function returns a list after converting every element of `A` into a string. The `join` function *joins* the elements of that list and returns a string in which the string elements of the sequence have been joined by a string separator (`''`). All that is left for us to do is to convert `s` to an integer type and add `1` to it. This solution is relatively simple, but you will not be asked for it in an interview.

Let's implement the algorithm that we discussed in the slides.

```

A1 = [1, 4, 9]
A2 = [9, 9, 9]

# s = ''.join(map(str, A))
# print(int(s) + 1)

def plus_one(A):
    A[-1] += 1
    for i in reversed(range(1, len(A))):
        if A[i] != 10:
            break
        A[i] = 0
        A[i-1] += 1
    if A[0] == 10:
        A[0] = 1
        A.append(0)
    return A

print(plus_one(A1))
print(plus_one(A2))

```



On **line 9**, we add **1** to the last element of the list **A**. Now we have to handle the carry as a result of this addition. Therefore, using a **for** loop on **line 10**, we iterate the list from the last index to the index 1 through the **reversed** function. Next, on **line 11**, we check if we get **10** as a result of adding **1**. If not, then the carry from the addition is 0, and we break the loop on **line 12**. However, if **A[i]** is **10** then we put **0** in its place and add **1** to the preceding index on **lines 13-14**. This is repeated for every position in the list from the last index to index 1.

Now, we need to handle one other edge case, i.e., the value at the first index is **10**. If **A[0]** is **10**, we set **A[0]** to **1** and append **0** to the list on **lines 16-17**. **A** is finally returned from the function on **line 18** after being incremented by 1.

Hope the above solution was easy to understand! Let's look at another problem regarding arrays in the next lesson.