Question

The interview question for this week is adding one to a list representation of a number: https://leetcode.com/problems/plus-one/

Potential solutions

Here's a potential solution that uses iteration. It starts by incrementing the ones digit. If that has overflown, then set the ones digit to 0 and add 1 to the tens digit, etc. Notice that in some cases (if the while loop completes), we need to add a new digit, as in the case when you add 1 to 999.

```python
def plusOne(digits):
    # start incrementing digits, starting
    # from the ones (rightmost) digit
    index = len(digits) - 1

    while index >= 0:
        digits[index] += 1
        if digits[index] < 10:
            # we don't need to overflow
            return digits
        else:
            # overflow to the next digit
            digits[index] = 0
            index -= 1

    # overflowed to the point where we
    # need a new digit, e.g. 999 + 1 = 1000
    return [1] + digits
```

Here's a potential solution that uses recursion:

```python
def plusOne(digits):
    if digits == []:
        # base case: we need to add an entirely new digit
        return [1]

    digits[-1] += 1
    if digits[-1] <= 9:
        # base case
        # we don't need to overflow to the next digit
        return digits

    # overflow to the next digit
    return plusOne(digits[:-1]) + [0]
```

And here’s a solution that converts the list of numbers to a string, then converts that string to an integer, adds one, and then reconverts it back into a list:

```
def plusOne(digits):
    # convert list to string [1, 2, 3] → ‘123’
    original_string = ''
    for d in digits:
        original_string += str(d)

    # convert string to int and add 1 to it
    num = int(original_string) + 1

    # convert the string back into a list of digits
    added_string = str(num)
    new_digits = []
    for s in added_string:
        new_digits.append(int(s))

    return new_digits
```

Follow-up questions

- **What’s the running time of your algorithm?**

  The answer is that all three of the above potential solutions run in linear time, since in the worst case when you add one to a number, you need to update every digit. So here, n = the number of digits.

- **Does your algorithm work if a new digit needs to be added? For example, if you add 1 to 999?**

  This case is easy to forget/skip over when writing the algorithm, so make sure that it is covered in the tests or in your questions.

- **Can you think of a different way of implementing the algorithm?**

  Encourage the interviewee to think about one of the other ways of implementing the algorithm. The three main ways are: iteration (using a loop), recursion, and converting the input to a number.