

37. Max Difference

You Can Get From Changing an Integer You are given an integer num. You will apply the following steps exactly two times: ● Pick a digit x ($0 \leq x \leq 9$). ● Pick another digit y ($0 \leq y \leq 9$). The digit y can be equal to x. ● Replace all the occurrences of x in the decimal representation of num by y. ● The new integer cannot have any leading zeros, also the new integer cannot be 0. Let a and b be the results of applying the operations to num the first and second times, respectively. Return the max difference between a and b. Example 1: Input: num = 555 Output: 888 Explanation: The first time pick x = 5 and y = 9 and store the new integer in a. The second time pick x = 5 and y = 1 and store the new integer in b. We have now a = 999 and b = 111 and max difference = 888 Example 2: Input: num = 9 Output: 8 Explanation: The first time pick x = 9 and y = 9 and store the new integer in a. The second time pick x = 9 and y = 1 and store the new integer in b. We have now a = 9 and b = 1 and max difference = 8

PROGRAM:

```
def maxDiff(num):
    num_str = str(num)
    max_val = num
    min_val = num
    for i, digit in enumerate(num_str):
        if digit != '9':
            max_val = int(num_str.replace(digit, '9'))
            break

    if num_str[0] != '1':
        min_val = int(num_str.replace(num_str[0], '1'))
    else:
        for i, digit in enumerate(num_str):
            if digit != '0' and digit != '1':
                min_val = int(num_str.replace(digit, '0'))
                break

    return max_val - min_val

num1 = 555
num2 = 9
```

```
print(maxDiff(num1))
```

```
print(maxDiff(num2))
```

```
888
```

```
8
```

OUTPUT:

TIME COMPLEXITY: $O(\log n)$