LeetCode: Count Numbers with Unique Digits

Junshan Chen

Problem

https://leetcode.com/problems/count-numbers-with-unique-digits/
Given a non-negative integer n, count all numbers with unique digits, x, where $0 \leq x \leq 10^n$.

Solution (C++)

Runtime: 0 ms
Memory Usage: 8 MB

```cpp
class Solution {
public:
    int countNumbersWithUniqueDigits(int n) {
        int count = 0;
        for (int i = 1; i <= n && i <= 10; i++){
            int temp = 9;
            for (int j = 1; j < i; j++){
                temp *= 10 - j;
            }
            count += temp;
        }
        count++;
        return count;
    }
};
```

Explanation

Since we are counting the numbers with unique digits within the range of $[0, 10^n]$ and 10^n would have multiple 0's, the numbers we are considering has at most n digits. Then we can separate the cases by the number of digits.

Let $i \in \{1, 2, \cdots, min\{n, 10\}\}$ to be the iterator. It worth to note that the numbers with unique digits can at most have 10 digits since they only get 10 possible choices $\{0, 1, \cdots, 9\}$. So we will also set the range of i to be not greater than 10. Within each loop , we count all numbers with i digits and all digits are unique. For each i, we then iterate through the digits by $j \in \{1, \cdots, i\}$. Note there are only 9 choices ($\{1, 2, \cdots, 9\}$) for the first digit since it can't be 0, we set a initial count $temp = 9$. Then for the j th digits, we count the number of choice the j^{th} digit got, which is $(10 - j)$, and scale *temp* by a factor of $(10 - j)$. After the loop, increase *count* by *temp*. Then the final result of *count* is what we want.

Complexity Analysis

Runtime Complexity: $O(1)$.
Since each iterator i is in range of $[0, 10]$ and each iterator j is in range of $[1, i-1] \subset [1, 10]$. Thus the running time of this algorithm is bounded above by 100 which is a constant. So the total runtime would be $O(1)$.

Space Complecity: $O(1)$.
Since we only need to record 4 integers, the space complexity is $O(1)$.