

```
1  /*
2  给定一个非负整数 n，计算各位数字都不同的数字 x 的个数，其中  $0 \leq x < 10^n$ 。
3
4  示例：
5
6  输入：2
7  输出：91
8  解释：答案应为除去 11,22,33,44,55,66,77,88,99 外，在 [0,100) 区间内的所有数字。
9
10 来源：力扣（LeetCode）
11 链接：https://leetcode-cn.com/problems/count-numbers-with-unique-digits
12 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
13  */
```

分析:

- 方法一:暴力回溯遍历法，确实可见指数的复杂度增长.
- 方法二:排列组合法($9 \times 9 \times 8 \dots$).
- 方法三:查表法，只有少数的有限种可能，不算投机取巧.

方法一:C++_暴力回溯遍历法

```
1  class Solution
2  {
3
4      private:
5          void helper(    int&      ret_val    ,
6                        int        cur_n      ,
7                        int        cur_idx     ,
8                        vector<int>& visited
9                    )
10         {
11             if(cur_idx == cur_n)
12             {
13                 ret_val++;
14                 return;
15             }
16
17             for( int i = 0 ; i < 10 ; i++) /**/
18             {
19                 if( (visited[i] == 1)
20                    ||(i == 0 && cur_idx == 0)// 去掉以0开头的
21                 )
22                 {
23                     continue;
24                 }
25                 else
26                 {
```

```

27         visited[i] = 1;
28         helper(ret_val, cur_n, cur_idx+1, visited);
29         visited[i] = 0;
30     }
31 }
32 }
33
34 public:
35     int countNumbersWithUniqueDigits(int n)
36     {
37         int ret_val = 0;
38
39         if(n < 0)
40         {
41             return ret_val;
42         }
43
44         ret_val = 1; /* n = 0 */
45
46         if(n > 10)
47         {
48             n = 10;
49         }
50
51         vector<int> visited(10, 0); /* 0~9 */
52         for(int i = 1; i <= n; i++)
53         {
54             helper(ret_val, i, 0, visited);
55         }
56
57         return ret_val;
58     }
59 };
60
61 /*
62 执行结果:
63 通过
64 显示详情
65 执行用时 :92 ms, 在所有 cpp 提交中击败了10.08% 的用户
66 内存消耗 :8.3 MB, 在所有 cpp 提交中击败了13.51%的用户
67 */

```

方法二:C++_排列组合

```

1  class Solution
2  {
3
4      public:
5          int countNumbersWithUniqueDigits(int n)
6          {
7              int ret_val = 0;
8              int temp = 0;
9              int arr[10] = {9, 9, 8, 7, 6, 5, 4, 3, 2, 1};
10             int i = 0;
11             int j = 0;
12

```

```

13         if(n < 0)
14         {
15             return ret_val;
16         }
17
18         if(n > 10)
19         {
20             n = 10;
21         }
22
23         ret_val = 1;
24
25         for( i = 1; i <= n ; i++)
26         {
27             temp = 1;
28             for(j = 0 ; j < i ; j++)
29             {
30                 temp *= arr[j];
31             }
32
33             ret_val += temp;
34         }
35
36
37         return ret_val;
38     }
39 };
40 /*
41 执行结果:
42 通过
43 显示详情
44 执行用时 :0 ms, 在所有 cpp 提交中击败了100.00% 的用户
45 内存消耗 :8.1 MB, 在所有 cpp 提交中击败了72.30%的用户
46 */
47

```

方法三:C++_查表法

```

1  class solution
2  {
3
4      public:
5          int countNumbersWithUniqueDigits(int n)
6          {
7              int ret_val = 0;
8              int arr[11] =
9              {1,10,91,739,5275,32491,168571,712891,2345851,5611771,8877691};
10             if(n>=0)
11             {
12                 if(n>=10)
13                 {
14                     n = 10;
15                 }
16                 ret_val = arr[n];
17             }
18         }
19     }
20 }

```

```
18
19         return ret_val;
20     }
21 };
22
23 /*
24 执行结果:
25 通过
26 显示详情
27 执行用时 :0 ms, 在所有 cpp 提交中击败了100.00% 的用户
28 内存消耗 :8 MB, 在所有 cpp 提交中击败了89.86%的用户
29 */
```

AlimyBreak
2019.12.08