

```

1  /*
2  你有一套活字字模 tiles，其中每个字模上都刻有一个字母 tiles[i]。返回你可以印出的非空字母序列的数目。
3  示例 1:
4  输入: "AAB"
5  输出: 8
6  解释: 可能的序列为 "A", "B", "AA", "AB", "BA", "AAB", "ABA", "BAA"。
7  示例 2:
8  输入: "AAABBC"
9  输出: 188
10 提示:
11     1 <= tiles.length <= 7
12     tiles 由大写英文字母组成
13 来源: 力扣 (LeetCode)
14 链接: https://leetcode-cn.com/problems/letter-tile-possibilities
15 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。
16 */

```

分析:

- 方法一:回溯法,由于要去重,需要把原始数组进行排序,然后记录下递归调用helper的次数即可(注意去掉空集).
- 这里与[090 子集II](#)的处理不一样, "AB"与"BA"非同一组合,我们需要用*visited*数组来做标记.

方法一:C++\_回溯法

```

1  class Solution
2  {
3      private:
4
5          void helper(    int&                ret_val    ,
6                        vector<int>&          vi          ,
7                        vector<int>&          visited
8                    )
9          {
10
11              ret_val++;
12              for (int i = 0; i < vi.size();i++)
13              {
14                  if (visited[i] == 1)
15                  {
16                      continue;
17                  }
18                  if(    i > 0
19                      && vi[i] == vi[i - 1]
20                      && visited[i-1] == 0
21                  )
22                  {
23                      continue;
24                  }
25                  visited[i] = 1;

```

```

26         helper(ret_val, vi, visited);
27         visited[i] = 0;
28     }
29 }
30
31 public:
32     int numTilePossibilities(string tiles)
33     {
34         if (tiles.size() <= 1)
35         {
36             return tiles.size();
37         }
38         vector<int> vi(tiles.size(), 0);
39         vector<int> visited(tiles.size(), 0);
40         int ret_val = 0 ;
41
42         for (int i = 0; i < tiles.size(); i++)
43         {
44             vi[i] = (int)(tiles[i] - 'A');
45         }
46
47         sort(vi.begin(), vi.end());
48         helper(ret_val, vi, visited);
49         return ret_val-1; /*去掉空集*/
50     }
51 };
52 /*
53 执行结果:
54 通过
55 显示详情
56 执行用时 :12 ms, 在所有 cpp 提交中击败了80.00% 的用户
57 内存消耗 :8.1 MB, 在所有 cpp 提交中击败了100.00%的用户
58 */

```