

```

1  /*
2  假设有从 1 到 N 的 N 个整数，如果从这 N 个数字中成功构造出一个数组，使得数组的第 i 位
   (1 <= i <= N) 满足如下两个条件中的一个，我们就称这个数组为一个优美的排列。条件：
3
4      第 i 位的数字能被 i 整除
5      i 能被第 i 位上的数字整除
6
7  现在给定一个整数 N，请问可以构造多少个优美的排列？
8
9  示例1：
10
11  输入：2
12  输出：2
13  解释：
14
15  第 1 个优美的排列是 [1, 2]：
16      第 1 个位置（i=1）上的数字是1，1能被 i（i=1）整除
17      第 2 个位置（i=2）上的数字是2，2能被 i（i=2）整除
18
19  第 2 个优美的排列是 [2, 1]：
20      第 1 个位置（i=1）上的数字是2，2能被 i（i=1）整除
21      第 2 个位置（i=2）上的数字是1，i（i=2）能被 1 整除
22
23  说明：
24
25      N 是一个正整数，并且不会超过15。
26
27  来源：力扣（LeetCode）
28  链接：https://leetcode-cn.com/problems/beautiful-arrangement
29  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。
30  */

```

分析:

- 回溯算法莽起.

方法一:C++_回溯算法

```

1  class Solution
2  {
3
4      private:
5
6
7      void helper (    int&          ret_num          ,
8                     int          N                  ,
9                     vector<bool>& vb                ,
10                    int          cur_level            )
11      {
12
13          if(cur_level > N)
14          {

```

```

15         ret_num++;
16         return;
17     }
18
19     for(int i = 1; i <= N ; i++)
20     {
21         if(vb[i]==true)
22         {
23             continue;
24         }
25         else
26         {
27
28             if(i%cur_level==0 || cur_level%i==0)
29             {
30                 vb[i] = true;
31                 helper(ret_num,N,vb,cur_level+1);
32                 vb[i] = false;
33             }
34         }
35     }
36
37 }
38
39 public:
40     int countArrangement(int N)
41     {
42         if(N<1)
43         {
44             return 0;
45         }
46
47         vector<bool> vb(N+1,false);
48         int ret_num = 0;
49
50         helper(ret_num,N,vb,1);
51         return ret_num;
52
53
54     }
55 };
56
57 /*
58 执行结果:
59 通过
60 显示详情
61 执行用时 :256 ms, 在所有 cpp 提交中击败了43.78% 的用户
62 内存消耗 :8.2 MB, 在所有 cpp 提交中击败了72.22%的用户
63 */

```