

```

1  /*
2  给定两个整数 n 和 k，返回 1 ... n 中所有可能的 k 个数的组合。
3
4  示例：
5
6  输入：n = 4, k = 2
7  输出：
8  [
9      [2,4],
10     [3,4],
11     [2,3],
12     [1,2],
13     [1,3],
14     [1,4],
15 ]
16
17 来源：力扣（LeetCode）
18 链接：https://leetcode-cn.com/problems/combinations
19 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。
20 */

```

分析:

沿用二进制法

- 先行筛选二进制位1的个数k的组合.
- 一边统计一边筛选.

方法一:C++_二进制法先筛选再组合

```

1  class Solution
2  {
3      private:
4          int numOfbit1(unsigned int d)
5          {
6              int count=0;
7              for(; d; d &= d - 1)
8              {
9                  count++;
10             }
11             return count;
12         }
13     public:
14         vector<vector<int>> combine(int n, int k)
15         {
16             vector<int>      ans
17             vector<int>      temp
18             vector<vector<int>> ret_val
19             int              size_pow    =    pow(2,n)
20             unsigned int      i          =    0
21             unsigned int      j          =    0
22
23             for(i = 0; i < n ;i++)
24             {

```

```

25         ans.push_back(i+1);
26     }
27
28     for( i = 0 ; i < size_pow ; i++)
29     {
30         if(numOfbit1(i)==k)
31         {
32             temp.clear();
33             for( j = 0; j < n ; j++)
34             {
35                 if( i & ( 0x01 << j))
36                 {
37                     temp.push_back(ans[j]);
38                 }
39             }
40             ret_val.push_back(temp);
41         }
42     }
43     return ret_val;
44 }
45 };
46
47 /*
48 执行结果:
49 通过
50 显示详情
51 执行用时 :88 ms, 在所有 C++ 提交中击败了98.18% 的用户
52 内存消耗 :11.9 MB, 在所有 C++ 提交中击败了67.91%的用户
53 */

```

方法二:C++_先组合在筛选

```

1  class Solution
2  {
3      public:
4          vector<vector<int>> combine(int n, int k)
5          {
6              vector<int>      ans                ;
7              vector<int>      temp                ;
8              vector<vector<int>> ret_val          ;
9              int              size_pow    =    pow(2,n) ;
10             unsigned int      i          =    0      ;
11             unsigned int      j          =    0      ;
12
13             for( i = 1; i <= n ;i++)
14             {
15                 ans.push_back(i);
16             }
17
18             for( i = 0 ; i < size_pow ; i++)
19             {
20                 temp.clear();
21                 for( j = 0; j < n ; j++)
22                 {
23                     if( i & ( 0x01 << j))
24                     {

```

```
25         temp.push_back(ans[j]);
26     }
27 }
28
29     if(temp.size() == k )
30     {
31         ret_val.push_back(temp);
32     }
33 }
34
35     return ret_val;
36 }
37 };
38
39 /*
40 执行结果:
41 通过
42 显示详情
43 执行用时 :204 ms, 在所有 C++ 提交中击败了33.93% 的用户
44 内存消耗 :11.8 MB, 在所有 C++ 提交中击败了70.26%的用户
45 */
```

AlimyBreak
2019.10.04