

给定两个整数 n 和 k ，返回 $1 \dots n$ 中所有可能的 k 个数的组合。

示例:

```
输入: n = 4, k = 2
输出:
[
  [2,4],
  [3,4],
  [2,3],
  [1,2],
  [1,3],
  [1,4],
]
```

一般采用回溯法:

```
1 # 回溯法
2 #     result = []
3 #     def backtrack(路径, 选择列表):
4 #         if 满足结束条件:
5 #             result.add(路径)
6 #             return
7 #         for 选择 in 选择列表:
8 #             做选择
9 #             backtrack(路径, 选择列表)
10 #            撤销选择
```

```
1 def combine(self, n, k):
2     """
3     :type n: int
4     :type k: int
5     :rtype: List[List[int]]
6     """
7     def backtrack(first = 1, curr = []):
8         # 如果有k个数, 那么就满足约束条件
9         if len(curr) == k:
10             output.append(curr[:])
11         # 从 1到n遍历
12         for i in range(first, n+1):
13             # 第一步把 i 添加到 curr 做选择
14             curr.append(i)
15             # 第递归把 i+1 添加到 curr, backtrack(路径, 选择列表)
16             backtrack(i+1, curr)
17         # 撤销选择
```

```
18         curr.pop()
19     output = []
20     backtrack()
21     return output
```

方法二：直接用库函数

```
1 def combine(self, n, k):
2     nums = []
3     res = []
4     for i in range(1, n+1):
5         nums.append(i)
6     for tmp in itertools.combinations(nums, k):
7         res.append(tmp)
8     return res
```