

0077. 组合

👤 [ITCharge](#) ⌚ 小于 1 分钟

- 标签：回溯
- 难度：中等

题目链接

- [0077. 组合 - 力扣](#)

题目大意

给定两个整数 n 和 k ，返回范围 $[1, n]$ 中所有可能的 k 个数的组合。可以按任何顺序返回答案。

解题思路

组合问题通常可以用回溯算法来解决。定义两个数组 `res`、`path`。`res` 用来存放最终答案，`path` 用来存放当前符合条件的一个结果。再使用一个变量 `start_index` 来表示从哪一个数开始遍历。

定义回溯方法，`start_index = 1` 开始进行回溯。

- 如果 `path` 数组的长度等于 k ，则将 `path` 中的元素加入到 `res` 数组中。
- 然后对 $[start_index, n]$ 范围内的数进行遍历取值。
 - 将当前元素 i 加入 `path` 数组。
 - 递归遍历 $[start_index, n]$ 上的数。
 - 将遍历的 i 元素进行回退。
- 最终返回 `res` 数组。

代码

py

```
class Solution:
    res = []
    path = []
    def backtrack(self, n: int, k: int, start_index: int):
        if len(self.path) == k:
            self.res.append(self.path[:])
            return
        for i in range(start_index, n - (k - len(self.path)) + 2):
            self.path.append(i)
            self.backtrack(n, k, i + 1)
            self.path.pop()

    def combine(self, n: int, k: int) -> List[List[int]]:
        self.res.clear()
        self.path.clear()
        self.backtrack(n, k, 1)
        return self.res
```