

问题 3 编程实现

采用 python 语言实现：

1.主要算法部分：

```
def combine(n, k):  
    def backtrack(start, path):  
        if len(path) == k:  
            res.append(path.copy())  
            return  
        for i in range(start, n + 1):  
            path.append(i)  
            backtrack(i + 1, path)  
            path.pop()  
  
    res = []  
    backtrack(1, [])  
    return res
```

2.算法说明：

算法主要采用回溯法，用递归的方式生成各种组合

从 1 开始，逐步选择数字，直到组合长度为 k，然后回溯继续搜索其他可能性。

1.终止条件

if len(path) == k: res.append(path.copy()) return

当当前组合长度等于 k 时，保存组合并返回。

2.选择与递归

for i in range(start, n + 1): path.append(i) # 选择当前数字 backtrack(i + 1, path) # 递归处理

后续数字 path.pop() # 撤销选择（回溯）

先将 i 加入当前组合。

再从 i+1 开始继续选择（避免重复）。

移除 i 以尝试其他可能性。

3.截图：

用例 1：

```
请输入 n 的值（输入 0 退出）：4
请输入 k 的值：2
所有可能的组合为：[[1, 2], [1, 3], [1, 4], [2, 3], [2, 4], [3, 4]]
```

用例 2:

```
请输入 n 的值（输入 0 退出）：1
请输入 k 的值：1
所有可能的组合为：[[1]]
```

用例 3:

```
请输入 n 的值（输入 0 退出）：5
请输入 k 的值：3
所有可能的组合为：[[1, 2, 3], [1, 2, 4], [1, 2, 5], [1, 3, 4], [1, 3, 5], [1, 4, 5], [2, 3, 4], [2, 3, 5], [2, 4, 5], [3, 4, 5]]
请输入 n 的值（输入 0 退出）：█
```