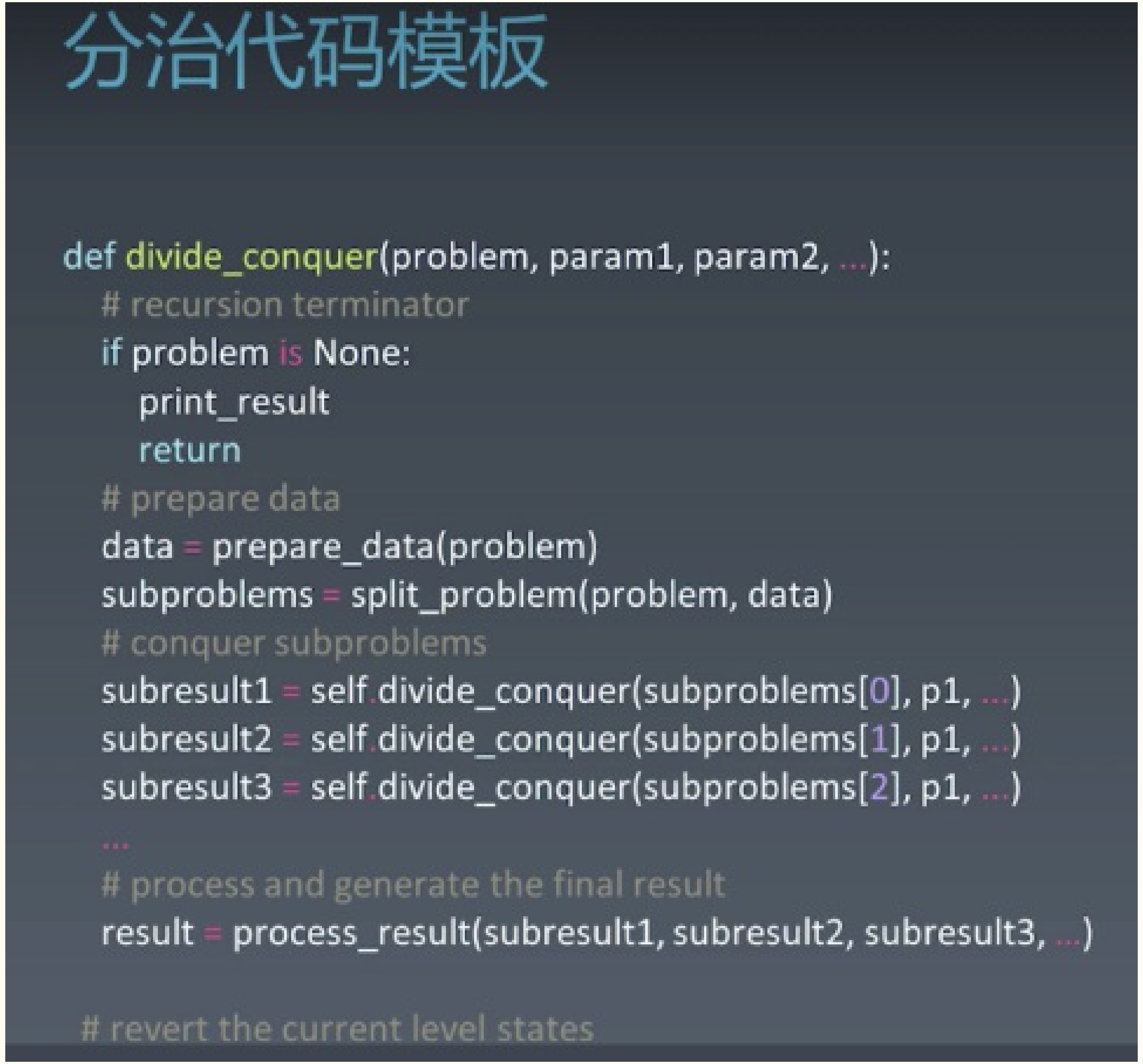

分治代码模板

```
def divide_conquer(problem, param1, param2, ...):
  # recursion terminator
  if problem is None:
    print_result
    return
  # prepare data
  data = prepare_data(problem)
  subproblems = split_problem(problem, data)
  # conquer subproblems
  subresult1 = self.divide_conquer(subproblems[0], p1, ...)
  subresult2 = self.divide_conquer(subproblems[1], p1, ...)
  subresult3 = self.divide_conquer(subproblems[2], p1, ...)
  ...
  # process and generate the final result
  result = process_result(subresult1, subresult2, subresult3, ...)

 # revert the current level states
```

分治

大问题由细的子问题组成

找最近重复性，将大问题分为子问题分别解决，再组合子问题的结果为最终结果

分治比泛型递归多一步

1. terminator (子问题没有了)
2. process(split your big problem)
3. drill down(sub problem)
4. merge(subsult)
5. reverse states

回溯

https://leetcode-cn.com/problems/permutations/solution/hui-su-suan-fa-xiang-jie-by-labuladong-2/

就是不断的下浅到每一层去试，找到结果

试完了再回到上一层去试其他的可能性