1.题型范式：

1.1回溯算法其实是一种穷举的搜索方法，利用回溯解决问题就是在穷举所有的可能，然后找到我们想要的答案，从一个搜索空间中找到满足条件的子搜索空间 1.2 常见关键字有排列问题，组合问题，子集问题等。

2.解题模板：

这类题目，最重要的就是根据题目要求，自己先动手在草稿纸上模拟一下可能的搜索流程，着重考虑如下的三个问题：

    (1) 可选搜索空间：所有可作出的搜索选择；

    (2) 已选搜索空间：也就是之前已经做出的搜索选择；

    (3) 结束条件：当已选搜索空间满足题目条件，可以结束穷举；

进一步地，需要考虑剪枝技术以减少无效地搜索空间

vector<elemType> res = {}

void backtrace(可搜索空间（假设题目给的是一个数组，通常通过一个索引来代表可选的范围，例如index=1，就代表接下来可以在1到length的范围中搜索）, 已搜索得结果，约束){

if(到达约束条件){

res.push\_back(已搜索得列表);

return;

}

for(当前层级可搜索的值的范围){

剪枝

If 满足约束：

做搜索选择;

#深一层级搜索（搜索空间中下一个位置的搜索）

backtrace(可搜索列表, 已搜索得结果);

撤销搜索选择;

}

}

3.常见变式：

3.1.类似于N皇后问题中，backTrack函数中的输入变量需要多几个以满足约束要求

3.2.类似于求组合总和问题中，数字可不可以重复。

例题1：

[39. 组合总和 - 力扣（Leetcode）](https://leetcode.cn/problems/combination-sum/description/)

例题2：

[40. 组合总和 II - 力扣（Leetcode）](https://leetcode.cn/problems/combination-sum-ii/description/)

例题3：

[216. 组合总和 III - 力扣（Leetcode）](https://leetcode.cn/problems/combination-sum-iii/description/)  
例题4：

[51. N 皇后 - 力扣（Leetcode）](https://leetcode.cn/problems/n-queens/description/)

例题5：

[37. 解数独 - 力扣（Leetcode）](https://leetcode.cn/problems/sudoku-solver/description/)