

回溯算法：（递归）

1、解决回溯问题，实际上就是一个**决策树**（因为在每一个节点上面都是在做决策）的遍历过程,纯暴力穷举

算法的核心问题有三个：

- 1、路径：就是已经做出的选择
- 2、选择列表：当前可以做出的选择
- 3、结束条件：到达决策树底层，无法再做选择的条件

在框架基础上要改的代码有：前序遍历部分和后序遍历部分要添加的操作。

```
1 result=[]
2 def backtrack(路径,选择列表):
3     if 满足结束条件:
4         result.add(路径);
5         return;
6
7     for 选择 in 选择列表:
8         做选择,从选择列表中拿出(移除)一个选择放入路径
9         backtrack(路径,选择列表)
10        撤销选择,从路径中拿出(移除)一个选择放入选择列表
11    注:
12    做选择是前序遍历,就是进入某一个节点之前的那个时间点执行
13    撤销选择是后序遍历,就是离开某个节点之后的那个时间点执行
14
```

2、全排列问题：

```
1 List<List<Integer>> res=new LinkedList<>;
2 List<List<Integer>> permute(int[] nums){
3     //记录路径
4     LinkedList<Integer> track=new LinkedList<Integer>();//track只是一个临时的容器,
5     //当到达结束条件时会重新创建一个列表.
6     backtrack(nums,track);
7     return res;
8 }
9
10 void backtrack(nums,track){
11     if(track.size()==nums.size()){
12         res.add(new LinkedList<Integer>(track));
13         return ;
14     }
```

```

15  for(int i=0;i<nums.size();i++){
16  //排除不合理的选择
17  if(track.contain(nums[i])){
18  continue;
19  }
20  track.add(nums[i]);//做出选择
21  backtrack(nums,track);
22  track.removeLast();//撤销选择
23  }
24  }

```

3、N皇后问题

```

1  vector<vector<int>> res;
2  vector<vector<int>> solveNQueens(int n){
3  int[][] board=new int[n][n]();
4  vector<int> ans=new vector<int>;
5  backtrack(board,ans,n,0);
6  return res;
7  }
8
9  void backtrack(board,ans,n,row){
10  if(row==n){
11  res.add(ans);
12  return;
13  }
14
15  for(int col=0;col<n;col++){
16  if(!isValid(board,row,col))continue;
17  board[row][col]=1;//表示放置皇后
18  ans.add(col);
19  backtrack(board,ans,n,row+1);
20  ans.remove();
21  board[row][col]=0;//表示撤销皇后
22  }
23  }
24
25  bool isValid(board,row,col){
26  for(int i=0;i<row;i++){
27  //判断是否有同一列
28  if(board[i][col]=='1') return false;

```

```
29 //判断board[row][col]左上角和右上角的是否有皇后
30 int leftcol=col-row+i;
31 int rightcol=col+row-i;
32 if(leftcol>=0&&board[row][leftcol]==1) return false;
33 if(rightcol<n&&board[row][rightcol]==1) return false;
34 }
35 }
```

如果只要找到一个答案即可

```
1 bool backtrack(路径, 选择列表){
2     if(满足条件){
3         return true;
4     }
5     for 选择 in 选择列表:
6         做选择
7         if(backtrack(路径,选择列表)) return true;
8         撤销选择
9     }
```