Algorithm 第二章初级排序算法

1 选择排序

首先找到最小值，与第一个元素交换，然后再找出剩下元素的最小值，与第二个元素交换，以此往复，直到有序。

2 插入排序

关键是从当前位置向前比较，如果大于，则被比较的往后移动。使用于部分有序或完全有序的情况（O（n））.

3 shell排序

基于插入排序的算法。通过把原数组变成间隔h有序的数组，然后进行插入排序，这样可以减少移动的次数（因为部分有序），关键是递增数列的构造。如果没有系统函数提供，可以先实现shell排序，因为其简单。

4 merge sort

关键是先sort再merge，merge时先copy到temp数组，然后根据四个条件把temp数组的元素放回原数组中，注意for循环从lo到hi，而不是0到len-1.

5 快速排序

先partition在sort，partition时while终止条件为两指针相遇，可以不需要swap函数。关键是先从右边开始比较。

BinarySearch

1 正常情况

2 找第一个比target大的数

3 找第一个比target小的数

4 找一个数的上限或者下限，可以用一个函数解决，记得定义idx和isFirst

TwoPointer

一般用while比较好

1 头尾指针，向中间收缩

2 快慢指针，慢指针在快指针里面跑，一个方向，lc209

DP

如果发现题目没有什么比较好的数据结构或者算法可以解决，或发现结果有一定规律，可以尝试dp。

1 看懂题目，从基本情况往前推（lc343），查找规律，有时候得从后面往前推。

2 推测规律，规律可能有

* 当前最优解与上次的结果有关（lc152，lc276），上次的结果可能不唯一，所以要根据上次结果来决定当前结果，这也是为什么要用数组把前面的结果存起来，因为可能还与之前几次或者之前第几次的结果有关（lc343）

3 验算规律，常用的方法有找几个数值，把规律代进去，看看是不是。最好的就是数学归纳法，还有其他的一些数学证明，缺点是要求时间和数学功底。

4 常用技巧

* 用数组存之前的最优解。
* 用几个全局变量表示可能的几种情况，比如lc198中，小偷打劫这间房对全局变量maxRob和maxNotRob的影响（maxRob应为之前maxRob和之前maxNotRob + house[i]的最大值）

Backtracking

其实就是DFS，不过增加了恢复现场操作，使用于操查找所有路径或者集合（lc78）.

1 看懂题目，确定DFS的参数，一般DFS或者BFS都有visited矩阵，还有一些会随着DFS变化的参数，如pos，left，used，prefix等

2 把visited矩阵相应位置变成true。

3 修改DFS的参数，如pos，left，used，prefix

4 恢复现场，把visited变成false，把存了的值吐出来