P41

Graphical user interface, application

Description automatically generated with medium confidence

**交换。**将理应在在位置的元素放到该位置上。忽略不合适的元素。完成该操作后，从头便利所有元素，第一个元素i+1便是first Missing Positive。

必须考虑一种特殊情况，在交换的时候。需要考虑元素相同的情况。

Text

Description automatically generated

标记：

将正数移动到前边，负数移动到后边，当｜nums[i]｜=m时，标记nums[m-1]位置处的元素取负。遍历完原先所有的正数时，从0开始检测，当i位置元素为正数时，表示缺失i+1元素。

P39

Graphical user interface, text, application, email

Description automatically generated

深度优先搜索，每个元素可以重复使用，使用前缀和或者remain来保证当前list中的元素是否满足等于target。重复解出现的实质，相同的元素作为开始。

技巧总结，在写dfs时，最好在函数中包含开始元素，使用contains有诸多不妙。

Text

Description automatically generated

动态规划。 //对于等于5的可能的集合的计算是nums[i]+{ans\_sub(target-nums[i])}

Text

Description automatically generated

取出和为sum的ans\_sum，在原来的基础上添加元素。对于ans\_sub则是复制其元素，在原来的基础上加入nums[i]。

Text

Description automatically generated

P40

在P39基础上修改。

Dfs。Start编程i+1。

Graphical user interface, text

Description automatically generated

start=i+1保证每一个元素使用一次。If语句保证了不会出现之前元素构建的子结构。

但是不能保证得到的list是不重复的。

Text

Description automatically generated

移除了右边分支的元素和当前节点相同的情况。i>start && ( nums[i] == nums[i - 1])。

[1,2,2,2,5]

for (int i = start; i < nums.length; i++) {  
   
 if (i>start && ( nums[i] == nums[i - 1])) {  
 continue;  
 }

[[1, 2, 2, 2, 5], [1, 2, 2, 5], [1, 2, 5], [1, 5], [2, 2, 2, 5], [2, 2, 5], [2, 5], [5]]

for (int i = start; i < nums.length; i++) {  
 if (temp.size()==0 && ( i>0&&nums[i] == nums[i - 1])) {  
 continue;  
 }

[[1, 2, 2, 2, 5], [1, 2, 2, 5], [1, 2, 2, 5], [1, 2, 5], [1, 2, 2, 5], [1, 2, 5], [1, 2, 5], [1, 5], [2, 2, 2, 5], [2, 2, 5], [2, 2, 5], [2, 5], [5]]

Text

Description automatically generated

无条件限制时：

[[1, 2, 2, 2, 5], [1, 2, 2, 5], [1, 2, 2, 5], [1, 2, 5], [1, 2, 2, 5], [1, 2, 5], [1, 2, 5], [1, 5], [2, 2, 2, 5], [2, 2, 5], [2, 2, 5], [2, 5], [2, 2, 5], [2, 5], [2, 5], [5]]

P38

Recursion

Text

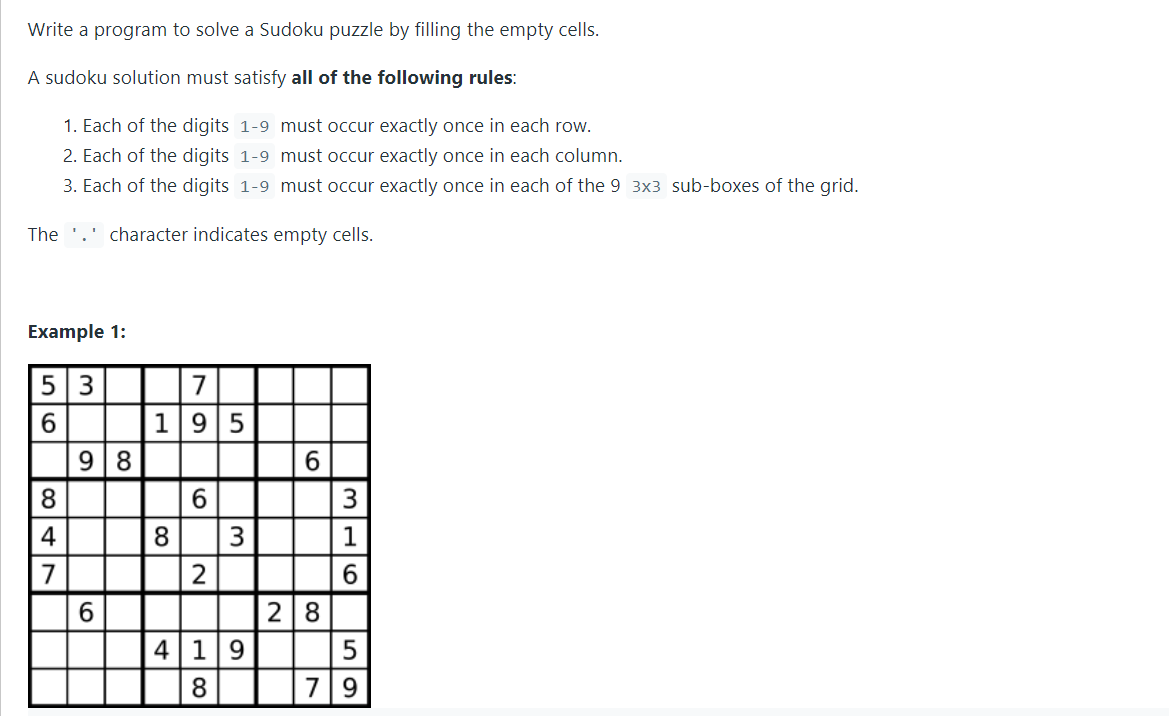
Description automatically generated

Iteration

Text

Description automatically generated

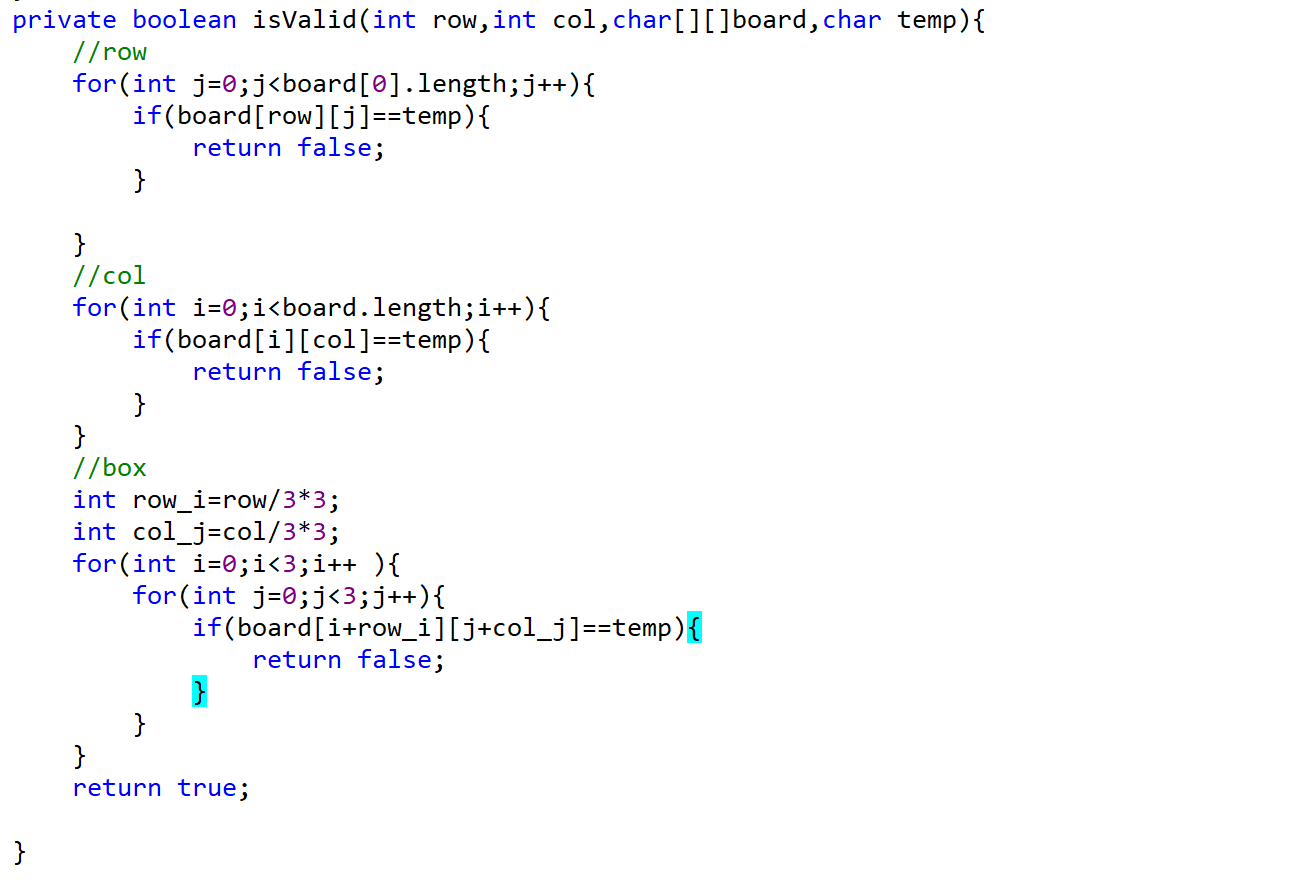
P37



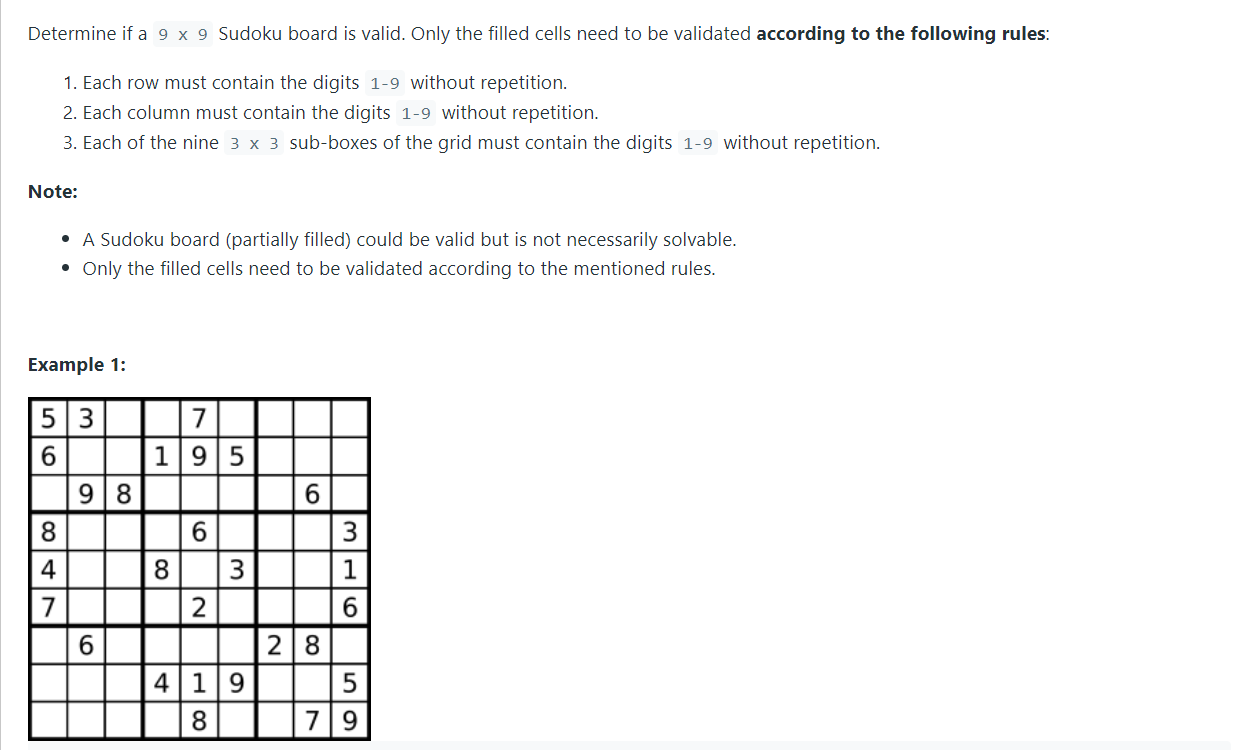
数独。先检查当前元素是否合规，然后带着当前元素继续回溯，一直到当前位置没有可用的数字添加或者到达最后一个元素。

检查盒子是否合规时，采用





P36



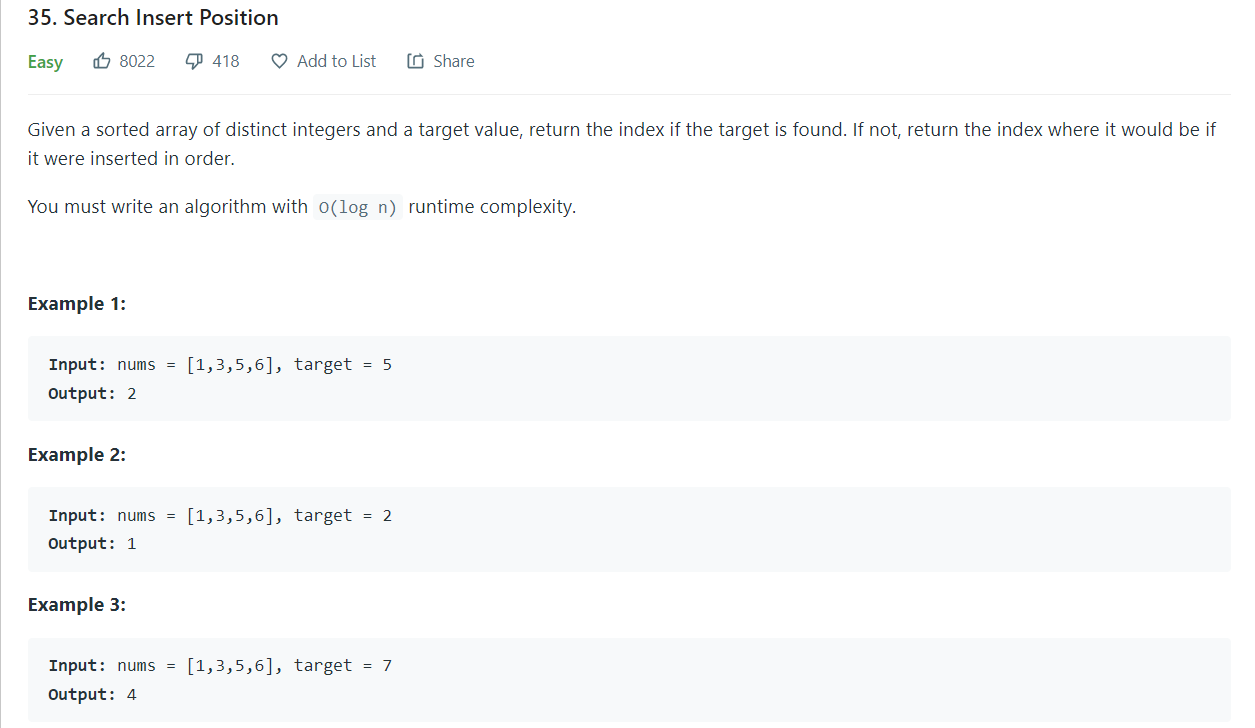
对表中的每一个数字进行标记。

char c=board[i][j];

if(!(set.add(c+"in the row"+i)&&set.add(c+"in the col"+j)&&set.add(c+"in the table"+i/3+j/3)))

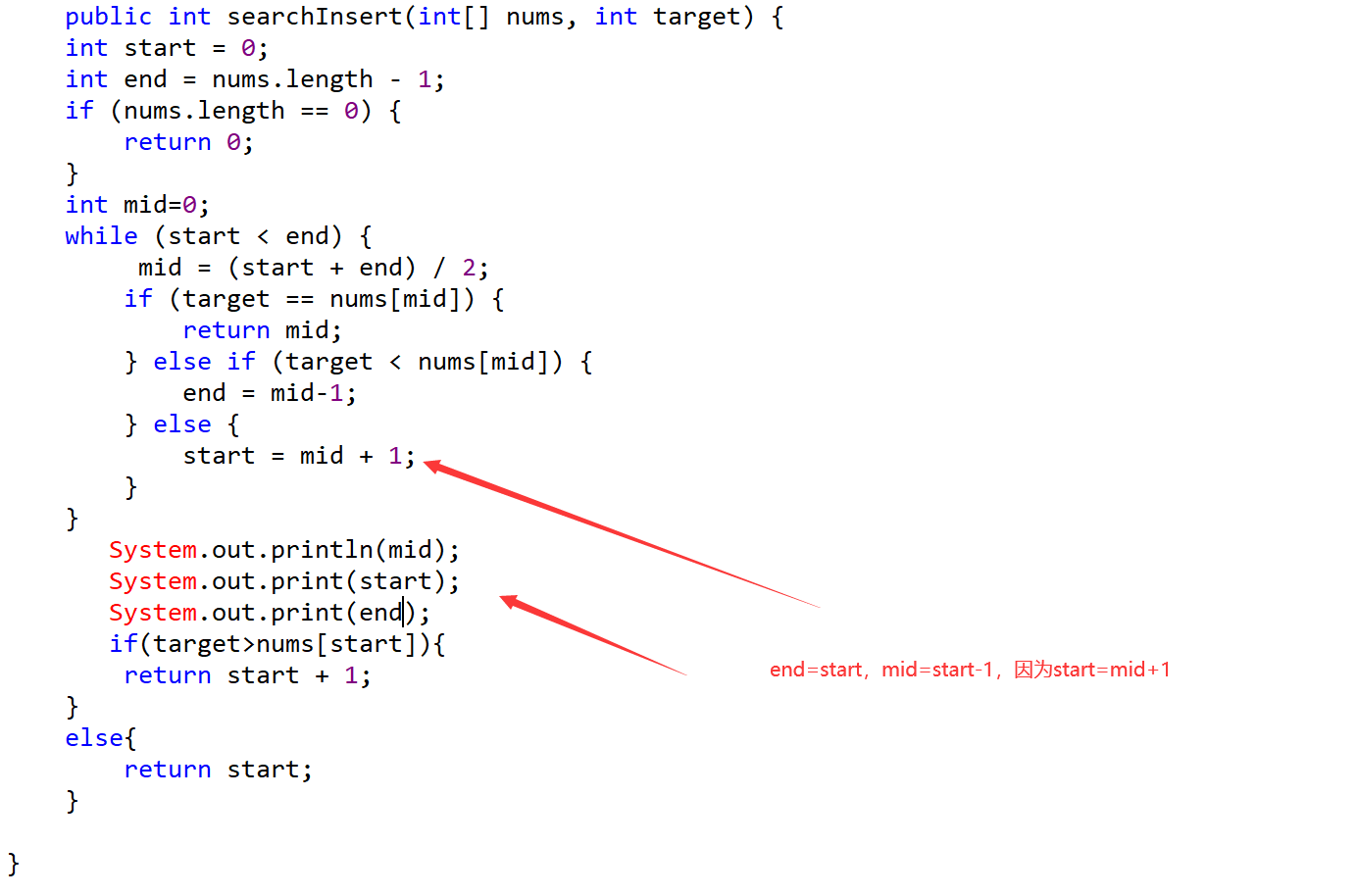
当该元素已经存在于行或者列，或者是grid时，返回false。否则返回true。

P35

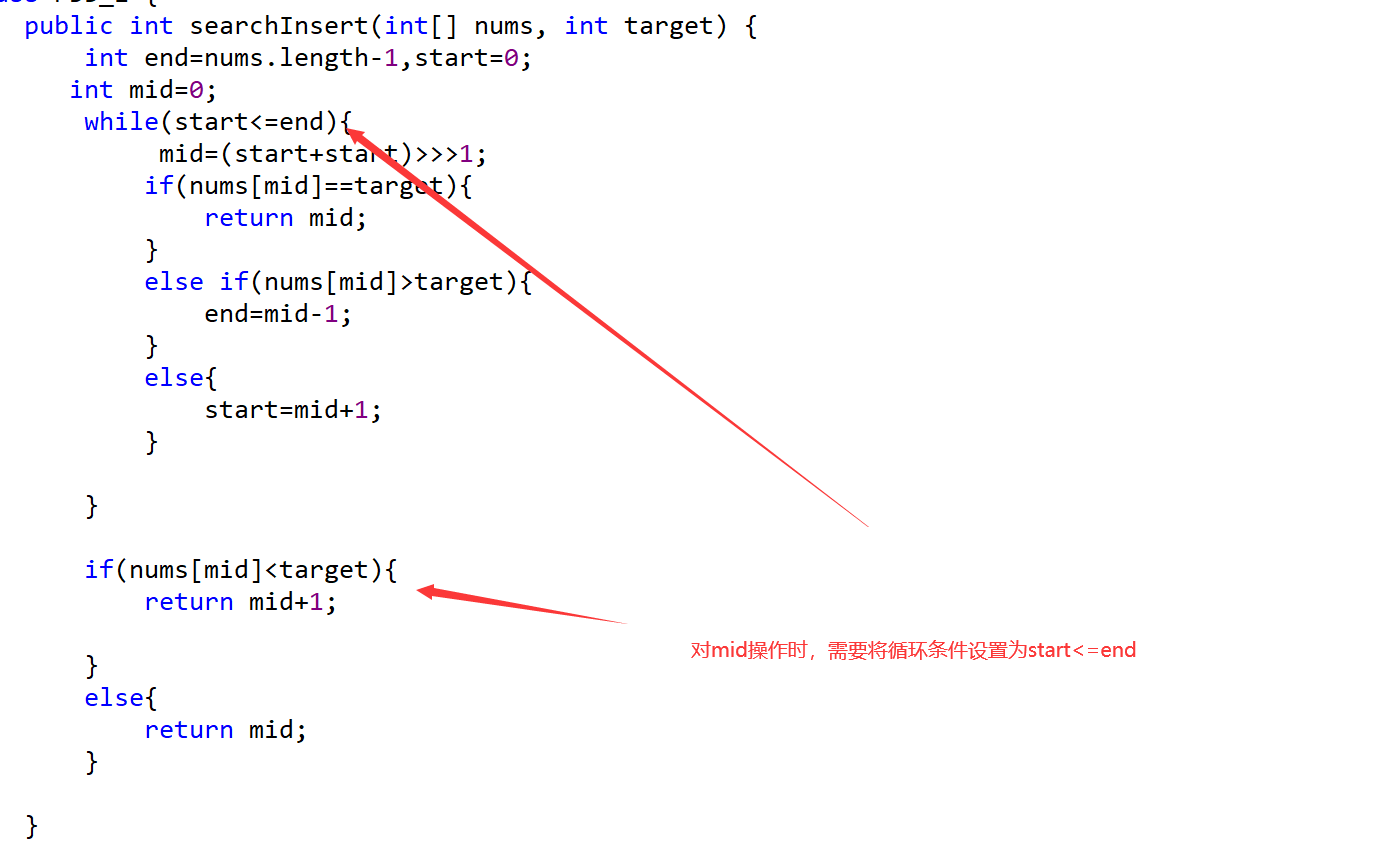


二分查找。

在二分查找中，需要将start=mid+1，end=mid-1。否则当存在偶数个元素时将陷入死循环，而且加一，减一操作跳过了重复元素。



当想要用mid作为输出时



P34



