算法讨论班第八期—李依霖

**2015-10-16**

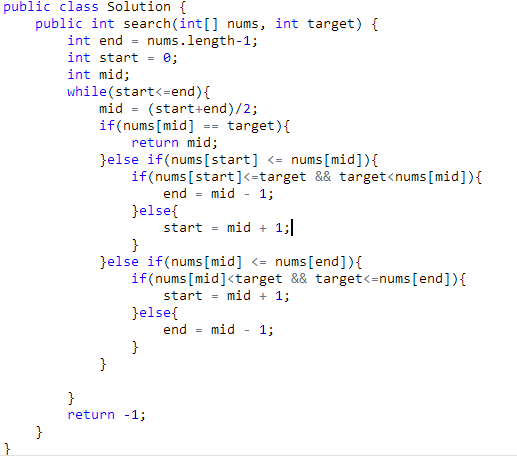
33. Suppose a sorted array is rotated at some pivot unknown to you beforehand.(i.e., 0 1 2 4 5 6 7 might become 4 5 6 7 0 1 2). You are given a target value to search. If found in the array return its index, otherwise return -1.

You may assume no duplicate exists in the array.

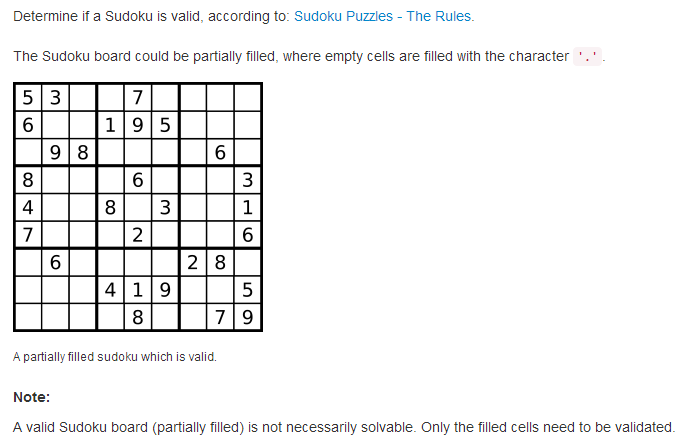
思路一：直接遍历一遍数组，找到对应数字对应的下标，时间复杂度为o(n).

思路二：找到数组中的转折点，然后分别在两部分进行二分查找，但是问题在于不知道最大值出现在数组中哪个位置，所以可以通过缩小查找范围逼近要查找的数字。

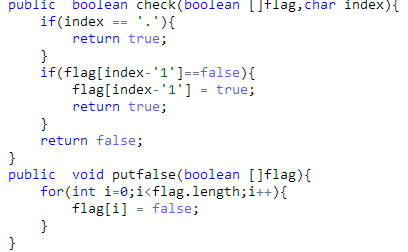
如果要查找的值恰好落在两个单调区间内，直接利用二分查找即可找到。如果没有，就通过缩小范围来逼急要查找的数字。



36.



思路：分别检查行、列和每九个格是否满足条件即可



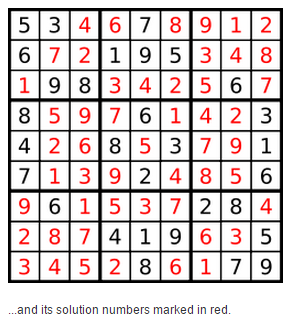
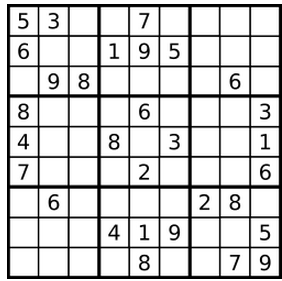
3.回溯法专题

1)Sudoku Solver

Write a program to solve a Sudoku puzzle by filling the empty cells

Empty cells are indicated by the character '.'.

You may assume that there will be only one unique solution.

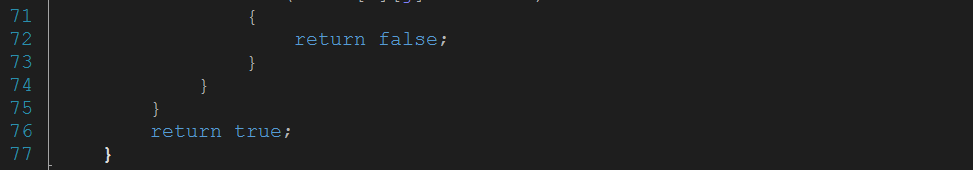
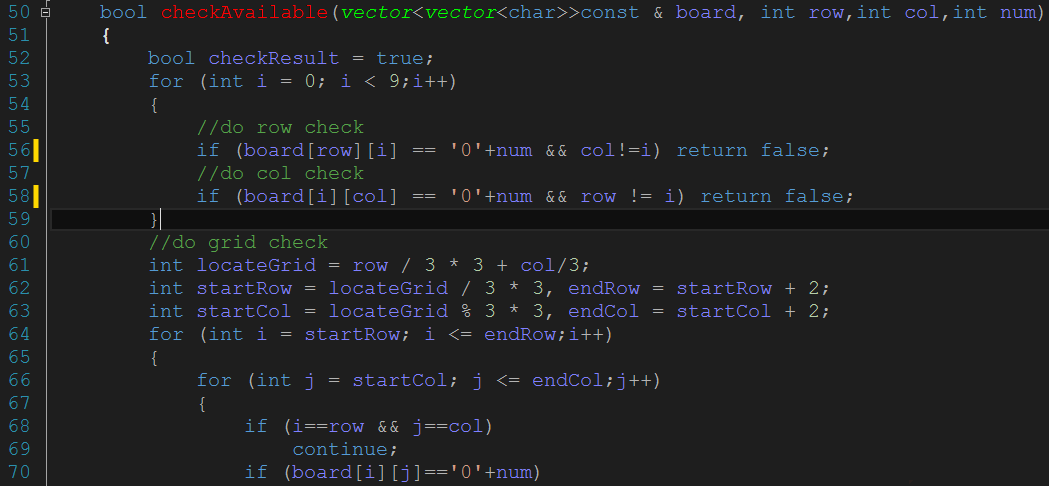


想想我们拿到一个数独题，正常人的思路应该是这样的:

* 遇到数字跳过，找到第一个空格
* 每个空格的数字都是1-9，随机填一个数，然后检查是否合适
* 如果有重复，就把当前的数字擦掉，然后选另外一个不同的数；重复上面的步骤直到找到一个可用数字填进去
* 相同的方法填下一个数字
* 如果当前数字已经没有数字可以选择，回退到上面一个数字，修改上面的数字，直到下面一个数字有填的——需要回溯！！！

似乎不难，只用解决两个问题：1）如何检查当前填的数字合适 2）如何回溯

问题1）在上一题我们已经涉及到了~



现在看问题2）：

回溯问题——第一想到用递归的方法去解决 为什么？——因为递归可以将临时变量入栈，当下次回溯时可以自动恢复现场

先试着尝试一下。。。问题在哪儿？？？

void solveSudoku(vector<vector<char>>&board)

* 表示棋盘的变量为非基本类型，非基本类型在c++中默认拷贝构造函数是浅拷贝，怎么解决？——修改STL,继承c++的vector类，重写父类的拷贝构造函数。——太麻烦了！！！

那怎么可以比较简单地保存现场？

首先明白在本题中，什么是现场？——1）当前棋盘中所有已经填过的数 2）当前已经填到哪了

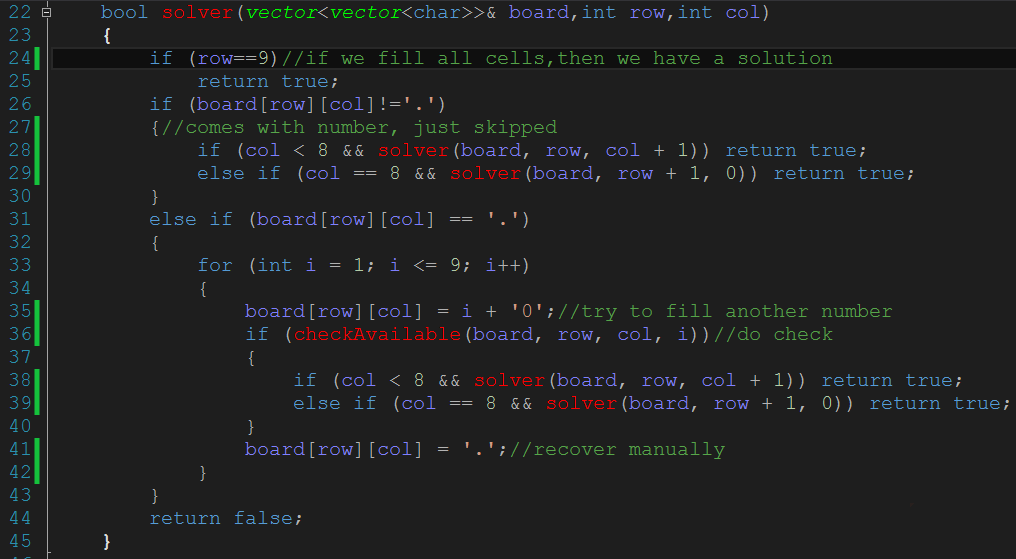
仔细思考一下，现场信息2）貌似可以用基本坐标——两个整形来表示，堆栈会自动保存，保证递归返回时可以恢复调用前的信息

但是基本信息1）呢？？？——手动恢复现场，如果填错了，就手动恢复现场。

怎么判断填错了？。。。当前是否填对，得看下一个能否有数填——递归的过程，直到最后一个有数填

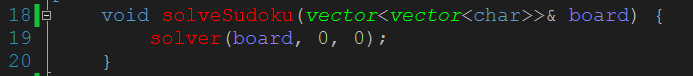
程序流程：

* 填数
* 检查通过-》填下一个空格(递归调用)
* 不通过-》换一个数填



思考：为什么返回bool ? -》只让求一个解

最后，主函数：

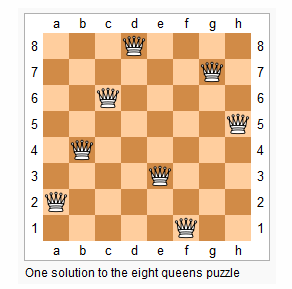


思考：

本题只让求一个解，如果想得到所有解怎么办？以下通过 N-Queens问题来讲解

1. [N-Queens](https://oj.leetcode.com/problems/n-queens/" \t "http://www.tuicool.com/articles/_blank)

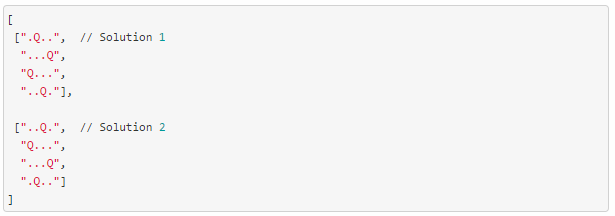
The n -queens puzzle is the problem of placing n queens on an n × n chessboard such that no two queens attack each other.



Given an integer n , return all distinct solutions to the n -queens puzzle.

Each solution contains a distinct board configuration of the n -queens' placement, where 'Q' and '.' both indicate a queen and an empty space respectively.

For example, There exist two distinct solutions to the 4-queens puzzle:



一般人解决问题：

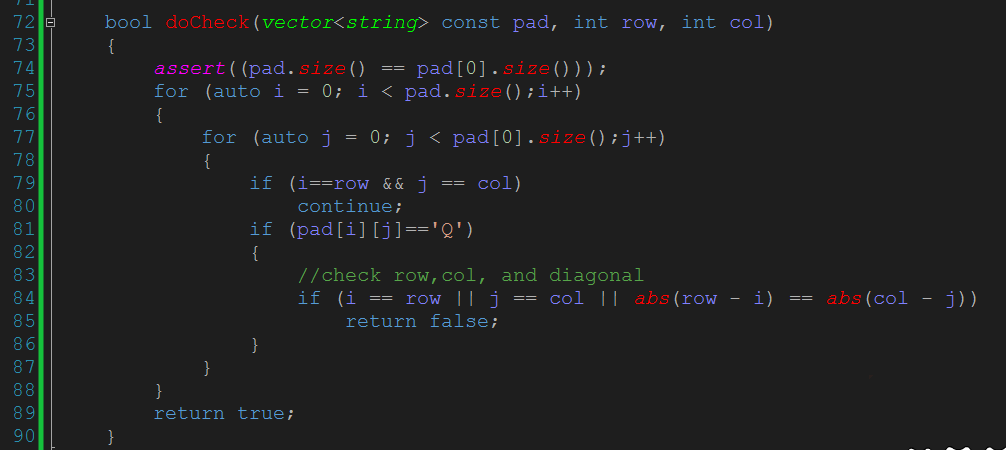
1. 找到空格，放一个Q
2. 一次往下找所有的空格，直到遇到合适的位置放下第二个Q

3）如果放到最后放不完Q,那么需要重新放前一个Q

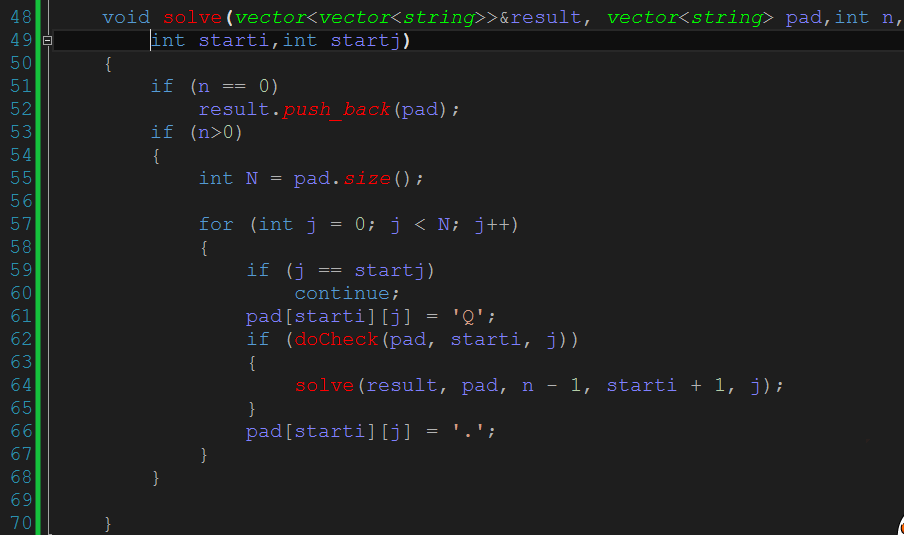
和前面的问题似曾相识？？？两个问题：

1）什么是合适？

2）如何回溯？



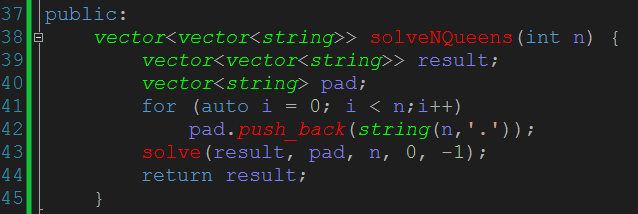
直接上代码：



用我们刚刚讲的，如何理解这段代码？？

Starti,startj如何理解？？——》用来加速算法

主函数：



具体代码，讨论班之后发.cpp文件和文档一起压缩后发QQ群~

能否有同学能帮忙总结一下，回溯问题的通用解法？？

下周由韩冬奇主讲，题目是：31,39,57