**目 录**

[1 基本数据结构 1](#_Toc93243055)

[2 二叉树 1](#_Toc93243056)

[3 回溯 1](#_Toc93243057)

[3.1 矩阵中的路径(单词搜索) 1](#_Toc93243058)

[3.2 机器人的运动范围 2](#_Toc93243059)

[4 动态规划 3](#_Toc93243060)

[4.1 剪绳子 3](#_Toc93243061)

[4.2 最长回文子串 4](#_Toc93243062)

[5 双指针 4](#_Toc93243063)

[5.1 三数之和 4](#_Toc93243064)

[6 滑动窗口 5](#_Toc93243065)

[6.1 解题框架 5](#_Toc93243066)

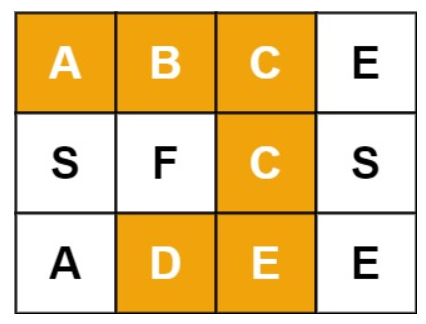
[6.2 无重复字符的最长子串 6](#_Toc93243067)

1. 基本数据结构
2. 二叉树
3. 回溯
   1. 矩阵中的路径(单词搜索)

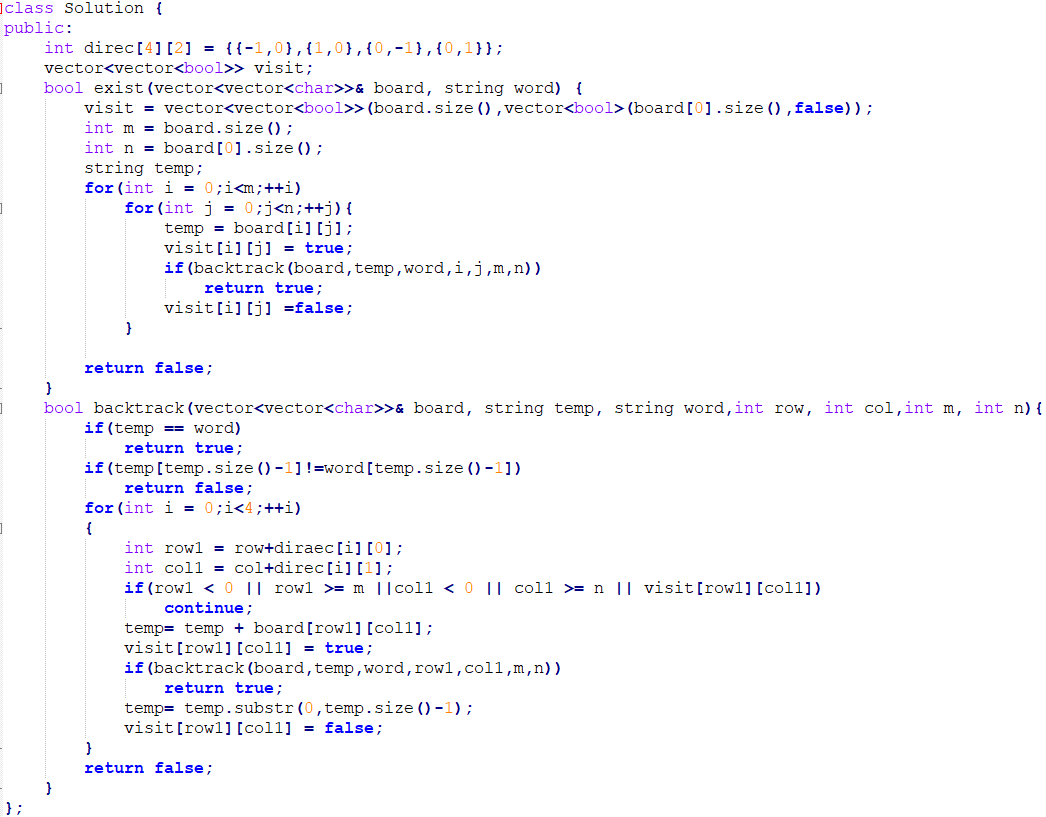
给定一个 m x n 二维字符网格 board 和一个字符串单词 word 。如果 word 存在于网格中，返回 true ；否则，返回 false 。

单词必须按照字母顺序，通过相邻的单元格内的字母构成，其中“相邻”单元格是那些水平相邻或垂直相邻的单元格。同一个单元格内的字母不允许被重复使用。

例如，在下面的 3×4 的矩阵中包含单词 "ABCCED"（单词中的字母已标出）



**【解题思路】**非常明显的回溯题，注意：1. 在主函数里面需要遍历二维数组获取单词起点，再进行回溯；2. 根据每一位字符进行剪枝，而不是每次判断整个单词是否相等。



* 1. 机器人的运动范围

地上有一个m行n列的方格，从坐标 [0,0] 到坐标 [m-1,n-1] 。一个机器人从坐标 [0, 0] 的格子开始移动，它每次可以向左、右、上、下移动一格（不能移动到方格外），也不能进入行坐标和列坐标的数位之和大于k的格子。例如，当k为18时，机器人能够进入方格 [35, 37] ，因为3+5+3+7=18。但它不能进入方格 [35, 38]，因为3+5+3+8=19。请问该机器人能够到达多少个格子？

**【解题思路】**回溯时需注意两个问题，1，递归不回溯，返回后visit不需要设回false；2，只需要向右向下移动即可，不需要遍历四个方向。



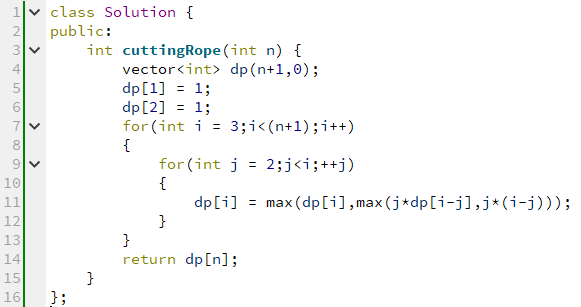
1. 动态规划
   1. 剪绳子

给你一根长度为 n 的绳子，请把绳子剪成整数长度的 m 段（m、n都是整数，n>1并且m>1），每段绳子的长度记为 k[0],k[1]...k[m-1] 。请问 k[0]\*k[1]\*...\*k[m-1] 可能的最大乘积是多少？例如，当绳子的长度是8时，我们把它剪成长度分别为2、3、3的三段，此时得到的最大乘积是18。

**【解题思路】**能想到是动态规划，取最大值时实际应该是：（共四项）因为m>1，其中(i-j)\*j表示只剪一刀，dp[i-j]\*dp[j]表示至少3刀，dp[i-j]\*j,和dp[j]\*[i-j]表示至少两刀。

max(dp[i-j]\*dp[j], (i-j)\*j, dp[i-j]\*j, dp[j]\*[i-j])

因为会进行迭代，所以有些像将是重复的，可以只保留其中两项。max(j\*dp[i-j],j\*(i-j))



* 1. 最长回文子串

给你一个字符串 s，找到 s 中最长的回文子串。

【解题思路】字符串子串的动态规划问题，需要定义2维dp数组，i,j分别表示子串的首位index。



1. 双指针
   1. 三数之和

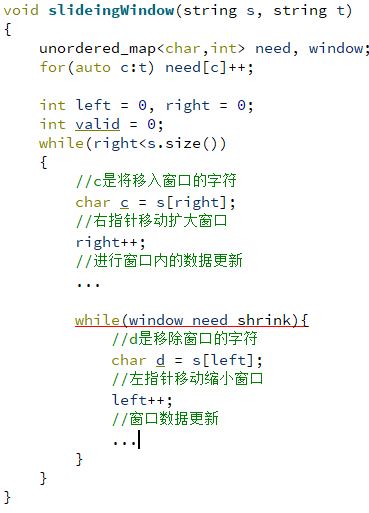
给你一个包含 n 个整数的数组 nums，判断 nums 中是否存在三个元素 a，b，c ，使得 a + b + c = 0 ？请你找出所有和为 0 且不重复的三元组。

注意：答案中不可以包含重复的三元组。

【解题思路】首先排序（排序后才能使用双指针？），遍历数组取第一个数，第二个数和第三个数采用双指针，题目的重点在于去重。



1. 滑动窗口
   1. 解题框架



* 1. 无重复字符的最长子串

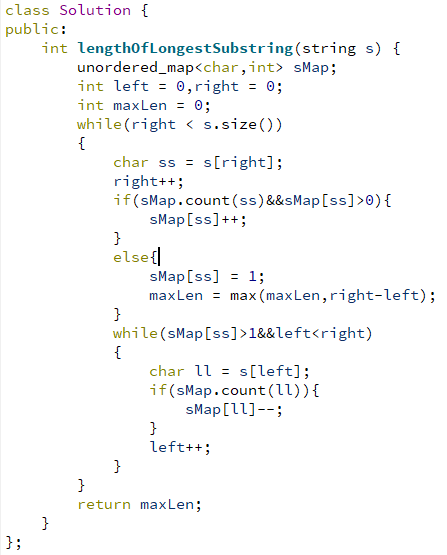
给定一个字符串 s ，请你找出其中不含有重复字符的 **最长子串**的长度。

**输入:** s = "abcabcbb"

**输出:** 3

**解释:** 因为无重复字符的最长子串是 "abc"，所以其长度为 3。

【解题思路】双指针，套模板



1. 快速幂
   1. 数值的整数次方

实现 [pow(x, n)](https://www.cplusplus.com/reference/valarray/pow/) ，即计算 x 的 n 次幂函数（即，xn）。不得使用库函数，同时不需要考虑大数问题。

【解题思路】快速幂法时间复杂度为log(n)，注意int 的溢出问题。

