



用递归思想、栈和队列解决问题

程序调用自身的编程技巧称为递归（recursion）。递归作为一种设计思想在程序设计中广泛应用。递归是一个过程或函数在其定义或说明中直接或间接调用自身的一种方法，它通常把一个大型复杂的问题层层转化为一个与原问题相似的规模较小的问题来求解。递归策略只需使用少量的程序代码就可描述出解题过程所需要的多次重复计算，大大地减少了程序的复杂性。递归的能力在于用有限的语句来定义对象的无限集合。一般来说，递归需要有边界条件、递归前进段和递归返回段。当边界条件不满足时，递归前进；当边界条件满足时，递归返回。

高级程序语言都支持递归函数的编写，从数据结构的角度可知，系统之所以可以支持递归调用，是因为有栈这种数据结构的存在。所以说，栈和递归调用是一对密不可分的好伙伴。

题目 1：判定一个字符串是否是回文串

回文串就是指不论正读还是反读都是一样的字符串。比如以下都是回文串：“radar”、“able was i ere saw elba”，另外，如果我们做一个可以忽略文本串中的空格的话，那么下面串也是回文串：“a man a plan a canal panama ”。

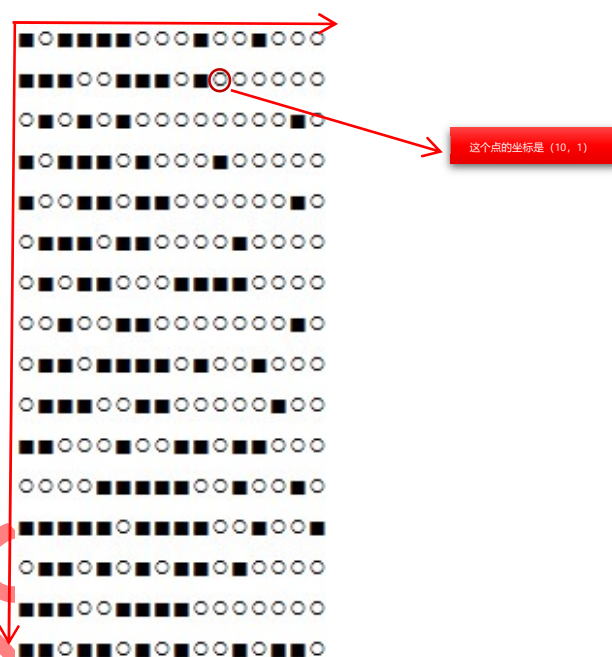
题目任务：

1. 要求使用递归思想，编写一个函数 `testPalindromeByRecursion`，该函数用来判断给定的字符串参数是否是回文串，如果是则返回 `true`，否则返回 `false`。
2. 要求使用栈数据结构，将 1 中编写的程序转换成非递归函数，函数名为：`testPalindromeByStack`，该函数用来判断给定的字符串是否是回文串，如果是则返回 `true`，否则返回 `false`。

注：本实验中使用的栈和队列都只能使用自定义的数据结构，以下同。

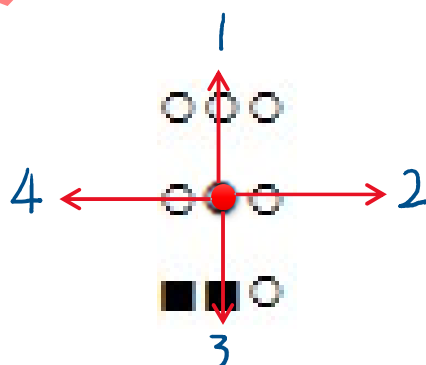
题目 2：走出迷宫

一个用 unicode 字符‘■’和‘○’构成的迷宫，将其存储在给定的输入文件中，其中‘■’代表墙壁，而‘○’代表迷宫中可以通行的道路。给定的迷宫都是一个正方形，其中迷宫中的墙壁或者道路都是



有坐标的，坐标的原点是整个迷宫的左上角，水平方向是 x 轴，垂直方向是 y 轴，如上图所示。

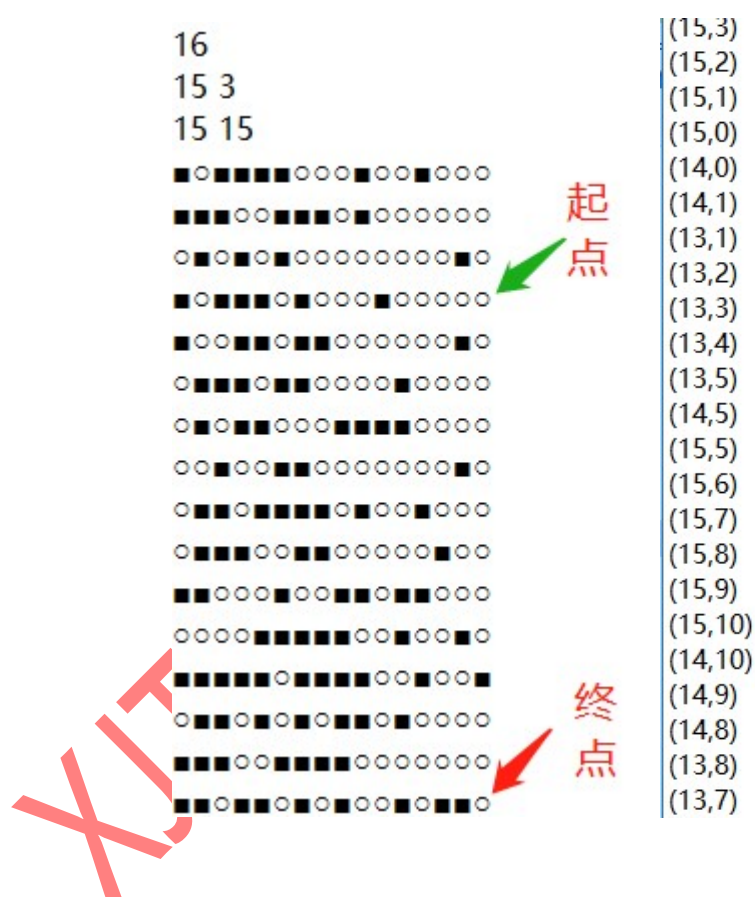
题目中给出探索这个迷宫的起点和终点坐标，要求能给出一个从起点到终点所构成的一个坐标点序列，该序列串起来就是所要求解的路径。为了保证路径的唯一性（不追求路径是最短），规定探索迷宫时的方向顺序为：上北、右东、下南和左西。具体示例如下：



也就是对于到达某个道路的一个坐标点时，先尝试探索北侧是否有路径可达，如果没有则考虑右侧是否有路径可达，依次类推。

实验为大家准备了四个用例，其中两个用例为测试用例(maze1.txt 和 maze2.txt，对应存放路径信息的文件是 result_maze1.txt 和 result_maze2.txt)，另外两个用例为检查作业正确性的用例。为了完成正确的测试，规定了输入和输出的数据格式，输入数据的格式主要是为了让我们知道如何读取迷宫的信息，而输出的格式是为了让我们判断所给定的路径是否是正确的路径。当四个测试用例都正确时，则说明你的迷宫探索程序是正确的。

下面是四个迷宫用例中的第一个用例数据，输入格式为：第一行是迷宫的大小；第二行是迷宫的入口点坐标（用空格分开两个值）；第三行是迷宫的终点坐标；剩余 16 行就是迷宫的方阵了。



上图中右侧的数据列是针对上面用例运行之后的输出数据（只有一部分），具体的格式为每一行都是一个用圆括号括起来的坐标对，第一行是迷宫的起点坐标，最后一行是迷宫的终点坐标，将这些坐标按顺序串起来就是迷宫的行走路径。（注意：圆括号和逗号都是英文符号，圆括号和数字之间是没有空格的。）

题目任务：

1. 要求使用递归思想，编写一个函数 `mazeTravelByRecursive`，该函数用来读取迷宫数据，然后输出正确地走迷宫的序列。
2. 要求使用栈数据结构，将 1 中编写的程序转换成非递归函数，函数名为：`mazeTravelByStack`，函数的功能如 1。

注：题目中用来代表墙和通道的 Unicode 字符的相应 Unicode 编码为：U+25A0 和 U+25CB。

题目 3：使用基数排序的思想完成排序

输入：一组整数类型的数据或者一组等长的字符串

输出：按照从小到大将整数类型的数据排好序，按照字典顺序将字符串数据排好序

说明：

■ 必须完成的要求

- ✓ 当数据序列是整数类型的数据的时候，数据序列中每个数据的位数不要求等宽，比如：



- ✧ 1、21、12、322、44、123、2312、765、56
- ✧ 请使用 radixSort1.txt 文件中的数据进行测试
- ✓ 当数据序列是字符串类型的数据的时候，数据序列中每个字符串都是等宽的，比如：
 - ✧ "abc","bde","fad","abd","bef","fdd","abe"
 - ✧ 请使用 radixSort2.txt 文件中的数据进行测试
- ✓ 当数据序列是字符串类型的数据的时候，如果数据序列中每个字符串不等宽，还能否使用基数排序思想进行排序？如果可以，请给出具体的排序思路 and 实现。
 - ✧ 请使用 radixSort3.txt 文件中的数据进行测试

XJTUSE