**难度: 简单**

输入一个矩阵，按照从外向里以顺时针的顺序依次打印出每一个数字。

示例 1：

输入：matrix = [[1,2,3],[4,5,6],[7,8,9]]

输出：[1,2,3,6,9,8,7,4,5]

示例 2：

输入：matrix = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]

输出：[1,2,3,4,8,12,11,10,9,5,6,7]

限制：

0 <= matrix.length <= 100

0 <= matrix[i].length <= 100

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/shun-shi-zhen-da-yin-ju-zhen-lcof

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路(来源于题解):**

1).源程序有注释

|  |
| --- |
| public int[] spiralOrder(int[][] matrix) {  if (matrix.length == 0) {  return new int[0];  }  int top = 0; //上  int bel = matrix.length - 1; //下  int left = 0; //左  int right = matrix[0].length -1; //右  //因为上面的长和宽都减了1所有这里需要加1  int[] result = new int[(bel + 1)\*(right + 1)];  int index = 0; //下标  while(true) {  //第一行 左边 移动 右边  for (int i = left; i <= right; i++) {  //top不变 i一直向右移 直到 到 边界  result[index++] = matrix[top][i];  }  //头向下移动,看看是否大于底了  if (++top > bel) {  break;  }  //此时由 上 往下走,  //到 底 就停止  for (int i = top; i <= bel; i++) {  //此时最右不变, 但是头要一直往下  result[index++] = matrix[i][right];  }  //此时最右边的所有都没有了  //判断左边是否大于最右边减1的  if (left > --right) {  break;  }  //需要将最下的右边 向 左边遍历  for (int i = right; i >= left; i--) {  //此时最下不变,右边向左移动  result[index++] = matrix[bel][i];  }  //此时最下面已经全部存储完了,  //把最下向上移动一行  if (top > --bel) {  break;  }  //需要将最下的左边一直向上遍历  for (int i = bel; i >= top; i--) {  //此时最下面要一直移动,但是最左不动  result[index++] = matrix[i][left];  }  //此时最左边的已经全部存储了,  //所有向右移动一行比较  if (++left > right) {  break;  }  }  return result;  } |