LeetCode #6

作者：friedwm
原文链接：https://ld246.com/article/1536589475845
来源网站：链滴

问题：z字形排列给定字符串，第一个字符在左上角，第二个字符串在第二行，直到打印了n行，接着上打印，直到字符串遍历完，按行顺序输出所有字符。

将字符串 "PAYPALISHIRING" 以Z字形排列成给定的行数：

P A H N

A P L S I I G

Y I R

输出：PAHNAPLSIIGYIR

思路1：对每个字符建模，记录应在的行和列，排序后输出

思路2：每行维护一个StringBuilder，逐个字符append到合适的行，最后遍历所有StringBuilder

```
package xyz.quxiao.play.lab.leetcode;

import com.google.common.collect.Lists;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;
import java.util.TreeMap;
import lombok.Data;

/**
 * zigzag * @author 作者 :quxiao 创建时间：2018/8/29 16:56
 */public class Problem6 {

  public static void main(String[] args) {
    String s = "PAYPALISHIRING";
    Problem6 problem6 = new Problem6();
    System.out.println(problem6.convert(s, 4).equals("PINALSIGYAHRPI"));
    System.out.println(problem6.convert(s, 1).equals("PAYPALISHIRING"));
  }

  // 思路：逐个字符计算row,col；起始为(0,0)且向下移动，当上一个的row=numRow时，下一个转为向上，且col+1,row-1；当last.row = 0时，转为向下
  public String convert(String s, int numRows) {
    List list = Lists.newArrayList();
    boolean down = true;
    Str last = null;
    for (int i = 0; i < s.length(); i++) {
      Str str = new Str();
      str.setC(s.charAt(i));
      if (last == null) {
        str.setRow(0);
        str.setCol(0);
      } else {
        if (last.getRow() == 0) {
          str.setRow(1);
          str.setCol(last.getCol() + 1);
          down = true;
        } else if (last.getRow() == numRows - 1) {
          str.setRow(last.getRow() - 1);
```

```
        str.setCol(last.getCol() + 1);
        down = false;
      } else {
        if (down) {
          str.setRow(last.getRow()+1);
          str.setCol(last.getCol());
        } else {
          str.setRow(last.getRow()-1);
          str.setCol(last.getCol());
        }
      }
    }

    last = str;
    list.add(str);
  }

  Collections.sort(list, new Comparator() {
    @Override
public int compare(Str o1, Str o2) {
      if (o1.getRow() != o2.getRow()) {
        return o1.getRow() - o2.getRow();
      } else {
        return o1.getCol() - o2.getCol();
      }
    }
  });

  StringBuilder sb = new StringBuilder();
  for (Str str : list) {
    sb.append(str.getC());
  }
  return sb.toString();
}

@Data
public static class Str {

  private int row;
  private int col;
  private char c;

  public int getRow() {
    return row;
  }

  public Str setRow(int row) {
    this.row = row;
    return this;
  }

  public int getCol() {
    return col;
  }
```

```
    public Str setCol(int col) {
      this.col = col;
      return this;
    }

    public char getC() {
      return c;
    }

    public Str setC(char c) {
      this.c = c;
      return this;
    }
  }
}
```