**Week 4 Day 2**

### 51: N-Queens I

### Review III

51-1 如果copy 复制同一段代码请一定记住改变量名 比如i改成j。。。

51-2 .不要写成,

51-3 画图分析坐标之间的关系！

### 52 N-Queens II

52-1 My solution is easy to change

### Week 4 Day 3

### 53 Maximum SubArray

53-1 求最大值 可能是dp做法

### 54 Spiral Matrix

### Review I

54-1 for loop 请看本题个人答案为标准请背诵

### 55 JumpGame

55-1 在whileloop 中不断调整curMax可以够到的值，是greedy的做法

### 56 Merge Intervals

### Review III

56-1 sol1:可以用最简单的iterator遍历treeMap。按照Key的从小到大顺序

56-2 sol2: treeMap 重要的函数：

ceilingEntry(K key) return a key-value mapping associated with the least key greater than or equal to the given key, or null if there is no such key

ceilingKey(K key) return the least key greater than or equal to the given key, or null if there is no such key

firstEntry() return the first(lowest) key currently in this map

56-3 Sol 3: 很好的 Collections.sort(collection name, new Comparator<ColectionGenericType>(){

@Override

public int compare(GenericType g1, GenericType g2)

{

}

});

### Week 4 Day 4:

### 57 Insert Interval

### Revisit III

57-1 在一段数组中插入。在开头前和结尾后的特殊情况单独考虑！！！中间有一个交汇的过程。要考虑如果一开始就交汇会怎么样，如果交汇结束后立刻程序停止会怎么样。请阅读我自写代码的含义！

### 58 Length Of Last Word

### 59 Spiral Matrix II

### 60 Permutation Sequence

### Review II

60-1 boolean digits = new Boolean[9]; 声明和定义类型要相同

60-2 为什么要k-- revisit！！

60-3 反复提取数组中的第m个元素 可以用arraylist，提取后删除提取元素，index会自动更新

60-4 阶乘如何用！ 用array[i]存储 array[i+1] = (i+1)\*array[i] !!!

**Week 5 Day 1**

### 61 Rotate List

### 62 Unique Paths

62-1 相比recursion解法，dp解法的好处是同一个点只计算检索一次。

### 63 Unique Paths II

### 64 Minimum Path Sum

### Review I

64-1 请看题目本身 因为已经提供了一个matrix 所以Leetcode solution 3 4 本身就提供了解决问题的方法。

### 65 Valid Number

### Review II

65-1 boolean 数组不需要initialize （default认为都是false）。但是一个boolean单独变量需要initialize

65-2 当可以使用hashSet 如果元素所有的可能性已知，可以使用boolean 变量。变量的个数就是所有可能hashSet取值的个数。

### Week 5 Day 2

### 66 Plus One

66-1 进位题 顺序！！！先取得进位next再对当前为数取余：先做 / 再做 %

66-2 plus one 加1可以认为是初始时next==1 。注意第一位和其他位的区别