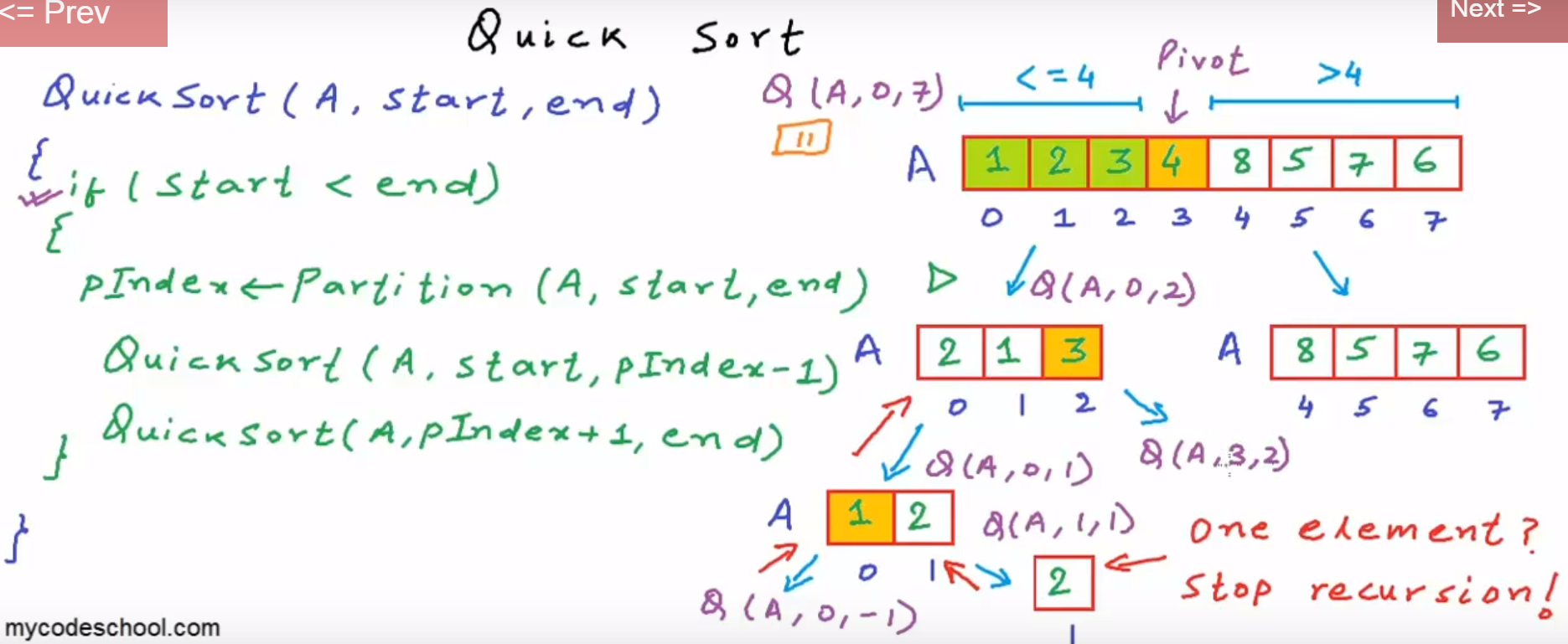
**Quick Sort快速排序**(use divide-and-conquer strategy 运用分解和递归思想)

1.找到一个中心值放到数组中间(pivot),将数组一分为二

2.用递归调用insertion sort的方法排序前半部分和后半部分

3.把两个分好的数组合并在一起



<https://www.youtube.com/watch?v=3Bbm3Prd5Fo>

Big Oh evaluate quick sort:

* In-place algorithm
* Time complexity:

O(n\*logn) => best or average case

O(n^2) =>worst case

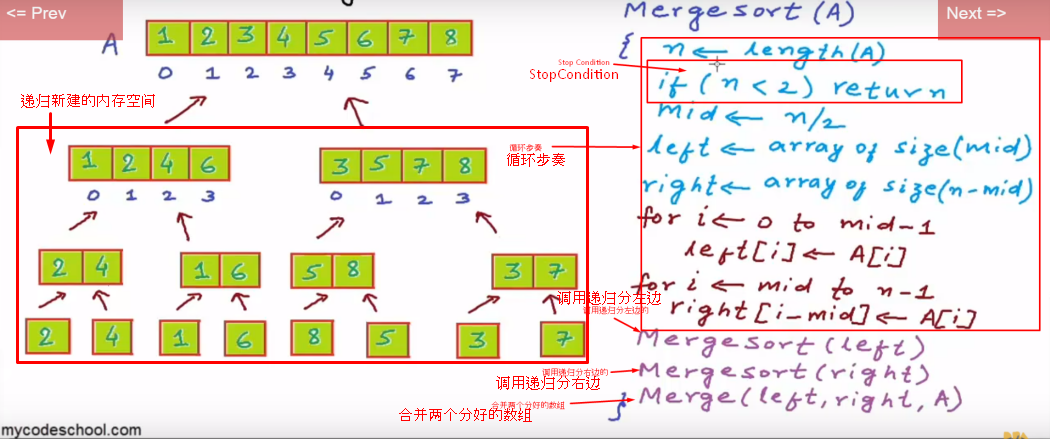
* not stable(if the array elements is pairs value(x,y))

**Merge Sort**(use divide-and-conquer strategy 运用分解和递归思想)

思想：

1.二分后递归调用

2.把1分好的两个数组合并到一起



优：Running time = O(n\*log2n)，运算速度成 log函数增长

劣：Space complexity=O(n) ,因为递归不断创建新数组会占用额外的内存空间(not in-place 不定的内存占用)

**Insertion sort 插入排序**

设置一个key值等于i,并每次外层循环 和已经分好的组里面元素比大小，比他小就替换(插进)前一个.

因为insertion sort用的是嵌套循环所以T=O(n^2),二次函数增长

代码:

public class InsertionSort {

public static int[] insertionSort(int[] list){

int i,j,key,temp;

for(i=1;i<list.length;i++) {

key=list[i];

j=i-1;

while(j>=0&&list[j]>key) {

temp=list[j];

list[j]=list[j+1];

list[j+1]=temp;

j--;

}

}

return list;

}

public static void main(String[] args) {

int[] ans= new int[]{3,4,5,6,7,1,4};

System.out.println(insertionSort(ans));

}

}

**排序是最重要的计算机操作之一**