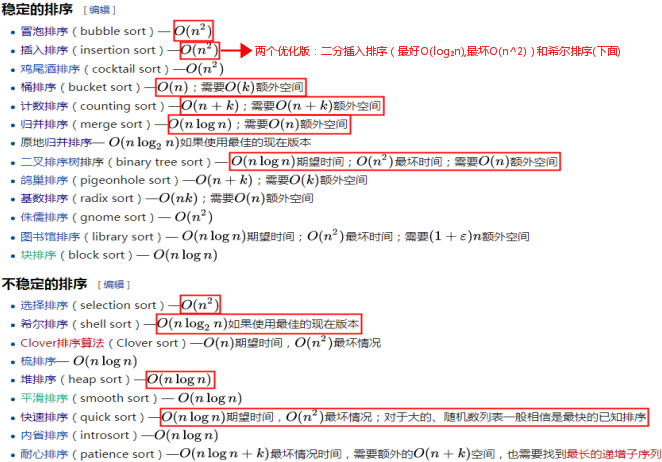
算法思想

* 时间复杂度
* 空间复杂度

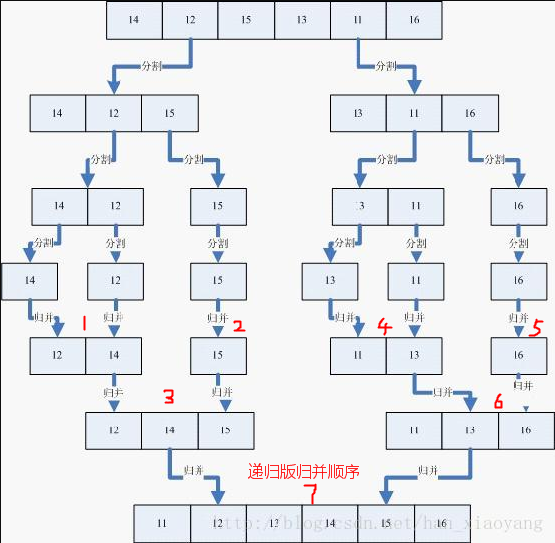
O(1) < O(logn) < O(n) < O(nlogn) < O(n^k) < O(n!) < O(n^n)

# 排序

## 常用的排序算法（维基百科）



注：归并排序理解图



## Java中的排序算法

### Arrays.sort()

如果数组长度大于等于286且连续性（连续升序和连续降序性）好的话，就用**归并排序**，如果大于等于286且连续性不好的话就用**双轴快速排序**。如果长度小于286且大于等于47的话就用**双轴快速排序**，如果长度小于47的话就用**插入排序**。

数据轴 -> **插入排序**【47】**双轴快速排序**【286】连续性好？**归并排序**：**双轴快速排序。**

### Collections.sort()

如果LegacyMergeSort.userRequested为true的话就会使用**归并排序**，可以通过代码设置为true（System.setProperty("java.util.Arrays.useLegacyMergeSort", "true");）如果不为true的话就会用一个叫**TimSort**的排序算法。

#### TimSort

<https://blog.csdn.net/yangzhongblog/article/details/8184707>

# 查找