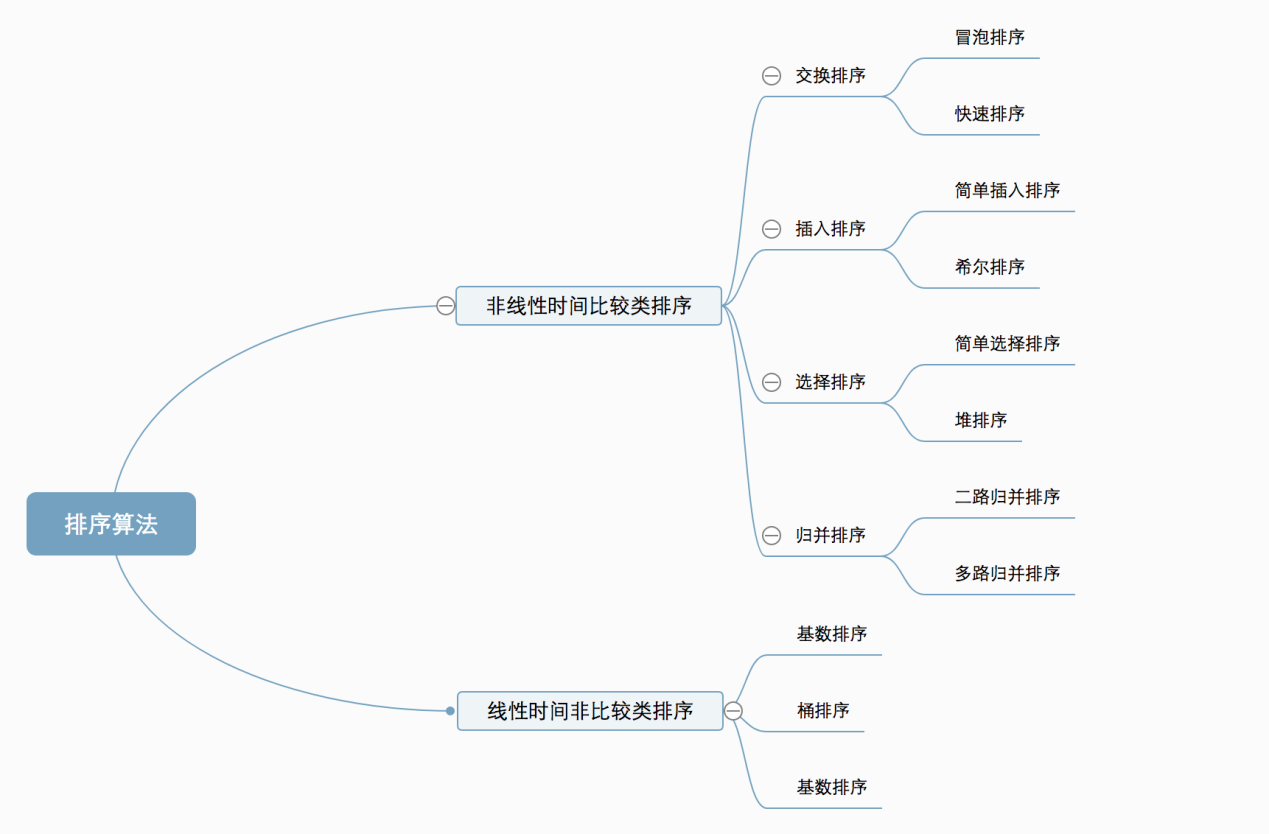
十大经典排序算法

1.分类

非线性时间比较类排序

线性时间非比较类排序



线性时间非比较类排序包括:计数排序,桶排序,基数排序

2.冒泡排序

属于一直比较相邻的两个,直到不交换

public void maopao(){

int[] str={1,3,44,21,445,652,234,2,4,5,123,54};

int strl=str.length;

for (int i = 0; i < str.length-1; i++) {

for (int j = 0; j < str.length-1-i; j++) {

if(str[j]>str[j+1]){

int temp = str[j+1]; // 元素交换

str[j+1] = str[j];

str[j] = temp;

}

}

}

System.out.println(Arrays.toString(str));

}

3.选择排序 寻找最大或者最小元素,取出,在剩余数组中继续取出最大或最小元素

public void xuanze(){

int[] str={1,3,44,21,445,652,234,2,4,5,123,54};

int mixIndex;

int temp;

for (int i = 0; i < str.length; i++) {

mixIndex=i;

for (int j = i+1; j < str.length; j++) {

if(str[j]<str[mixIndex]){

mixIndex=j;

}

}

temp=str[i];

str[i]=str[mixIndex];

str[mixIndex]=temp;

}

System.out.println(Arrays.toString(str)+"xuanze");

}

4.插入排序 选择当前元素,向前扫描直到找到比他小或者比他大的元素

public void charu(){

int[] str={1,3,44,21,445,652,234,2,4,5,123,54};

int preIndex;

int current;

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

preIndex=i-1;

current=str[i];

while(preIndex>=0&&str[preIndex]>current){

str[preIndex + 1] = str[preIndex];

preIndex--;

}

}

System.out.println(Arrays.toString(str)+"charu");

}

5.希尔排序 和插入排序相同,只是,会先与较远的元素进行比较

自己理解的

public void xier(){

int[] str={1,3,44,21,445,652,234,2,4,5,123,54};

int temp;

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

for (int j = 0; j < i; j++) {

if(str[i]<str[j]){

temp=str[i];

str[i]=str[j];

str[j]=temp;

}

}

}

System.out.println(Arrays.toString(str)+"xier");

}

复杂的:



6.归并排序 将数组分成n/2个子组,现将子组排序,再将几个子组组合,称为2-路归并.

写不出来



7.快速排序 通过一次排序将待排序记录分割成两部分,其中一部分记录的关键字逗比另一部分的小,分别对两部分排序,达到整体排序

选择一个基准值,比其小的都在左边,比其大的都在右边,递归形成



8.堆排序 使用二叉树的特性排序

9.计数排序

10.桶排序

11.基数排序,先按个位数排序,在按十位数排序

参考资料: https://www.cnblogs.com/onepixel/articles/7674659.html