第二章·算法入门

第一节·简单排序算法

1. ACM/ICPC算法预览。

在学习基础的算法之前，先让我们粗略的了解一下ACM/ ICPC之中常用的算法。见附录A。

简单排序算法，是ACM/ ICPC常用算法中的基础中的基础。下面，我们开始学习简单排序算法。这可能是你在ACM备赛中学习到的第一个算法系列。

1. 简单的排序算法简介

2.1什么是排序？

定义：

将杂乱无章的数据元素，通过一定的方法按关键字顺序排列的过程叫做排序。

这个定义，有两个中心点：**数据元素**，**关键字**。也就是说，一个排序操作，肯定含有关键字，和依照关键字排序的数据元素。找准这两个中心点，对我们进行排序非常重要。

2.2为什么排序？

2.2.1．为了更加快速的进行查找使用。

当我们处理总数为N的数据元素时，如果我们需要对其中的一个数据元素做处理，那么，我们首先要找到它。对于无序数据的查找，我们需要复杂度为0（N）的算法，即一个个的遍历所有的数据元素，直到找到我们需要的元素。但是，对于有序数据元素，算法的复杂度至少可以降低到0（logN）。

2.2.2．为了实战。

在ACM比赛中，所给的数据往往不是有序的，对所给数据做排序操作，是基本操守。

2.2.3为了好看，整洁。

当一堆杂乱的数据放在你的面前，如果不收拾好，不排个序啥的，怎么好意思。

2.3简单的排序算法简介

### **2.3.1** 插入排序(Insertion Sort)

基本思想：经过i-1遍处理后,L[1..i-1]己排好序。第i遍处理仅将L[i]插入L[1..i-1]的适当位置p，原来p后的元素一一向右移动一个位置, 使得L[1..i]又是排好序的序列。对于数据比较大的，通常可以采取二分查找来确定一个数应该加入的位置。

### **2.3.2** 希尔排序(Shell Sort)

基本思想：先取一个小于n的整数d1作为第一个增量，把文件的全部记录分成d1个组。所有距离为dl的倍数的记录放在同一个组中。先在各组内进行直接插入排序；然后，取第二个增量d2<d1重复上述的分组和排序，直至所取的增量dt=1(dt<dt-l<…<d2<d1)，即所有记录放在同一组中进行直接插入排序为止。

### **2.3.3** 冒泡排序(Bubble Sort)

基本思想是：对待排序的记录的关键字进行两两比较,如发现两个记录是反序的,则进行交换,直到无反序的记录为止。

### **2.3.4** 快速排序(Quick Sort)

### **2.3.5** 归并排序(Merge Sort)

归并就是将多个有序的数列合成一个有序的数列。将两个有序序列合并为一个有序序列叫二路归并(merge).归并排序就是n个长度为1的子序列,两两归并最后变为有序的序列。

1. **简单的排序算法理解与用法详解**

### **3.1** 插入排序(Insertion Sort)

我们在**2.3.1**中已经对插入排序有了个小小的了解。它的基本思想是不是有点难理解，其实换做大白话，就是：

对于一个数组元素总数为N的无序数组，我们假设，这个数组，由一个总数为N-1的**有序**数组和一个数A组成。那么，这个很好排序啊，把数据A，按照**关键字**排序，让A和N-1个数据一个一个比较，如果前面一个数据的关键字比A大，后一个数据的关键字比A小，然后把A放在它们中间，就完事了。

那么，回到现实，那个总数为N-1的其实是无序的数组。怎么办？

把N-1大小的数组看成是有一个N-2大小的有序数组和一个数据B组成。以此类推，到最后，我们剩下了总数为N-1的零散的数据（A,B,C,D等）和一个有序的数组X，虽然这个有序的数组只有一个数据元素。

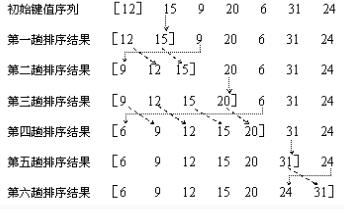
那么，把这个顺序倒过来，插入排序算法就实现了。请同学们根据我上面的大白话，对照下面的例子，好好理解。

例3-1：

依照数据的值从小到大编写程序运用插入排序算法实现：

数据：a[7]={12,15,9,20,6,31,24}

提示：

：

插入排序是线形排序，它是稳定的。

插入排序的时间复杂度最优为O(n),最坏为O(n^2),平均O(n^2）；

线形排序、归并排序的辅助空间为O(n),快速排序的辅助空间为O(logn),其它排序的辅助空间为O(1)

插入、冒泡排序的速度较慢，但参加排序的序列局部或整体有序时，这种排序能达到较快的速度。

附录A：

ACM/ICPC常用算法缩略

初级：

一.基本算法:    
    (1)枚举.   
    (2)贪心.   
    (3)递归和分治法.    
    (4)递推.    
    (5)构造法.  
    (6)模拟法.   
二.图算法:    
    (1)图的深度优先遍历和广度优先遍历.    
    (2)最短路径算法(dijkstra,bellman-ford,floyd,heap+dijkstra)   
    (3)最小生成树算法(prim,kruskal)   
    (4)拓扑排序 (poj1094)   
    (5)二分图的最大匹配 (匈牙利算法)   
    (6)最大流的增广路算法(KM算法).   
三.数据结构.    
    (1)串   
    (2)排序(快排、归并排(与逆序数有关)、堆排)   
    (3)简单并查集的应用.    
    (4)哈希表和二分查找等高效查找法(数的Hash,串的Hash)   
    (5)哈夫曼树   
    (6)堆    
    (7)trie树(静态建树、动态建树)   
四.简单搜索    
    (1)深度优先搜索   
    (2)广度优先搜索   
    (3)简单搜索技巧和剪枝   
五.动态规划    
    (1)背包问题.   
    (2)简单的深度优先搜索:    
      1.最长公共子序列       
      2.最优二分检索树问题    
六.数学    
    (1)组合数学:    
       1.加法原理和乘法原理.    
       2.排列组合.    
       3.递推关系.    
    (2)数论.    
       1.素数与整除问题    
       2.进制位.    
       3.同余模运算.   
    (3)计算方法.    
       1.二分法求解单调函数相关知识.   
七.计算几何学.    
    (1)几何公式.   
    (2)叉积和点积的运用(如线段相交的判定,点到线段的距离等).   
    (3)多边型的简单算法(求面积)和相关判定(点在多边型内,多边型是否相交)    
    (4)凸包.

中级：

一.基本算法:    
    (1)C++的标准模版库的应用.   
    (2)较为复杂的模拟题的训练   
二.图算法:    
    (1)差分约束系统的建立和求解.   
    (2)最小费用最大流   
    (3)双连通分量  
    (4)强连通分支及其缩点.   
    (5)图的割边和割点   
    (6)最小割模型、网络流规约   
三.数据结构.    
    (1)线段树.   
    (2)静态二叉检索树.   
    (3)树状树组   
    (4)RMQ.   
    (5)并查集的高级应用.   
    (6)KMP算法.   
四.搜索    
    (1)最优化剪枝和可行性剪枝    
    (2)搜索的技巧和优化   
    (3)记忆化搜索   
       
五.动态规划    
    (1)较为复杂的动态规划(如动态规划解特别的施行商问题等).  
    (2)记录状态的动态规划.   
    (3)树型动态规划   
六.数学    
    (1)组合数学:    
       1.容斥原理.    
       2.抽屉原理.    
       3.置换群与Polya定理    
       4.递推关系和母函数.    
          
    (2)数学.    
       1.高斯消元法   
       2.概率问题.   
       3.GCD、扩展的欧几里德(中国剩余定理)    
    (3)计算方法.    
       1.0/1分数规划.   
       2.三分法求解单峰(单谷)的极值.    
       3.矩阵法   
       4.迭代逼近   
    (4)随机化算法   
    (5)杂题.  
七.计算几何学.    
       (1)坐标离散化.    
       (2)扫描线算法(例如求矩形的面积和周长并,常和线段树或堆一起使用).    
       (3)多边形的内核(半平面交)   
       (4)几何工具的综合应用.

高级：  
一.图算法:    
     (1)度限制最小生成树和第K最短路.   
     (2)最短路,最小生成树,二分图,最大流问题的相关理论(主要是模型建立和求解)   
     (3)最优比率生成树.     
     (4)最小树形图  
     (5)次小生成树.    
     (6)无向图、有向图的最小环      
二.数据结构.     
     (1)trie图的建立和应用.   
     (2)LCA和RMQ问题(LCA(最近公共祖先问题) 有离线算法(并查集+dfs) 和 在线算法    
         (RMQ+dfs)).  
     (3)双端队列和它的应用(维护一个单调的队列,常常在动态规划中起到优化状态转移的   
         目的).    
     (4)左偏树(可合并堆).     
     (5)后缀树(非常有用的数据结构,也是赛区考题的热点).     
三.搜索     
     (1)广搜的状态优化:利用M进制数存储状态、转化为串用hash表判重、按位压缩存储状态、双向广搜、A\*算法.   
     (2)深搜的优化:尽量用位运算、一定要加剪枝、函数参数尽可能少、层数不易过大、可以考虑双向搜索或者是轮换搜索、IDA\*算法.   
四.动态规划     
     (1)需要用数据结构优化的动态规划.  
     (2)四边形不等式理论.    
     (3)较难的状态DP  
五.数学     
     (1)组合数学.    
       1.MoBius反演   
       2.偏序关系理论.    
     (2)博奕论.    
       1.极大极小过程  
       2.Nim问题.    
八.计算几何学.     
     (1)半平面求交   
     (2)可视图的建立  
     (3)点集最小圆覆盖.    
     (4)对踵点