

数据结构之堆排序

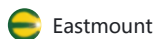
原创 Eastmount 2013-12-17 16:39:01 4570 收藏 7

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏,采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法、



¥9.90

订阅

在数据结构中,堆排序是非常重要的一个知识点,尤其在期末考试、考研等计算机考试中经常会考察堆排序,并要求画出示意图.下面主要通过一道考研题目讲述堆排序的知识,希望对大家有所帮助.

(文章内容参考严蔚敏的《数据结构》、王道论坛的《数据结构》和自己的一些理解)

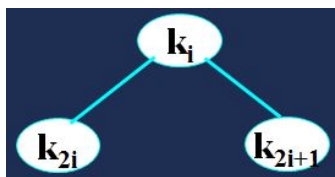
参看动态图: <http://www.benfrederickson.com/heap-visualization/>

一.堆排序定义

堆排序是一种**树形选择排序方法**,它的特点是在排序过程中,将 $L[1...n]$ 看成一颗完全二叉树的顺序存储结果,利用完全二叉树中双亲结点和孩子结点之间的内在关系,在当前无序区中选择关键字最大或最小的元素.该算法时间复杂度 $O(n\log 2n)$,是不稳定排序.

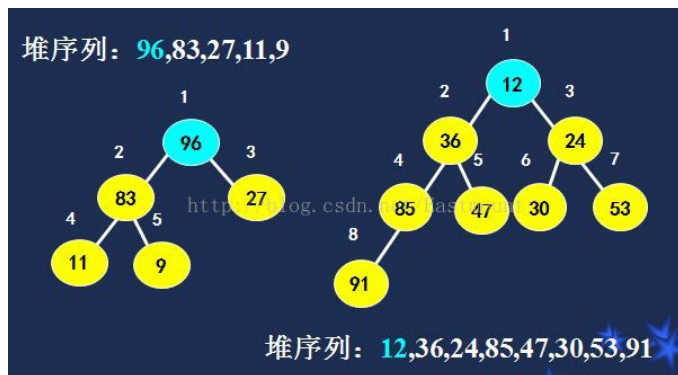
堆的定义: n 个元素的序列 $K[1..n]$ 当且仅当满足:

(1). $K[i] \leq K[2i]$ 且 $K[i] \leq K[2i+1]$ 或 (2). $K[i] \geq K[2i]$ 且 $K[i] \geq K[2i+1]$ ($i=1,2,...,\lfloor n/2 \rfloor$) 如下图所示:



满足(1)的堆称为小根堆,满足(2)的堆称为大根堆.

在大根堆中,堆顶元素(完全二叉树的根)必为序列中 n 个元素的最大值,且对其任一非根结点,它的值不大于其双亲结点值.对应的是最大堆.同理,小根堆定义相反,根节点为最小值.如下图分别是最大堆和最小堆.堆采用的存储方式是顺序存储结构.



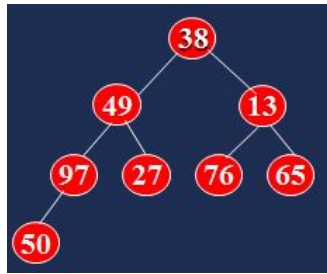
二.堆排序实现

堆排序利用堆的特性进行排序,其基本思想是:首先将待排序的记录序列构造成一个堆;然后选出对中有所有记录的最小值(最小堆);再把它从堆中输出,并把剩余的元素排成最小堆,依次找次小的记录并输出,直到只有一个记录为止.问题的步骤为:

- (1).如何从一个无序序列初始建堆,这是堆排序的关键.(初始建堆)
- (2).如何处理堆顶记录.(处理堆顶)
- (3).输出堆顶元素后如何调整剩余元素,构建一个新堆.(重新建堆)

例:(改考研题)使用堆排序对序列{38,49,13,97,27,76,65,50}进行排序,要求画出最小堆,并画出堆排序的示意图.

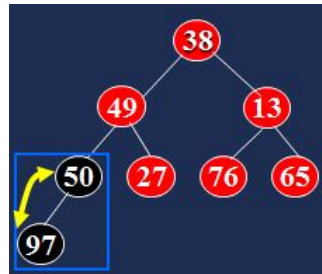
首先按照题目顺序构造一个完全二叉树如下图所示,然后按照下面的"初始建堆"和"堆排序"实现.



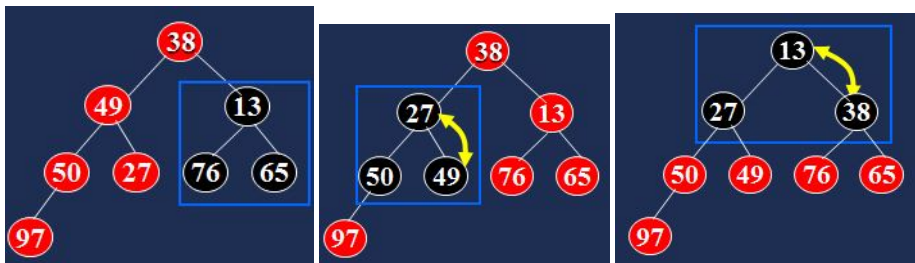
1.初始建堆

对初始序列建堆,就是有一个反复筛选的过程. n 个结点的完全二叉树,最后一个结点是第 $\lfloor n/2 \rfloor$ (其中 $\lfloor x \rfloor$ 表示不大于 x 的最大整数)个结点的孩子.小根堆其基本思想从 $\lfloor n/2 \rfloor$ 根节点开始,尽量让最小数向二叉树根结点(堆顶)移动,使子树都成为堆.

(1).对第 $\lfloor n/2 \rfloor$ 个结点为根的子树筛选,对于小根堆: 如果根结点的关键字大于左右子女结点中关键字较小者,则交换,使子树成为堆.题中: $i = \lfloor 8/2 \rfloor = 4$,指向97根结点,大于其左孩子结点50,故交换顺序.如下图.



(2).然后向前依次对各结点($\lfloor n/2 \rfloor - 1$ 到1)为根的子树进行筛选,看该结点值是否小于其左右子结点,如果不是,将左右子结点中较小者与之交换;交换后可能会破坏下一级的堆,继续采用上述方法构造下一级的堆,直到该结点为根的子树构成堆位置.



(3).反复使用上述方法建堆,直到根节点.题中根节点交换时亦要检查器下级堆是否要交换顺序.

2.堆排序

在使用完全二叉树初始建立好如上图的最小堆后,就是对该堆进行排序输出.此时的算法是:

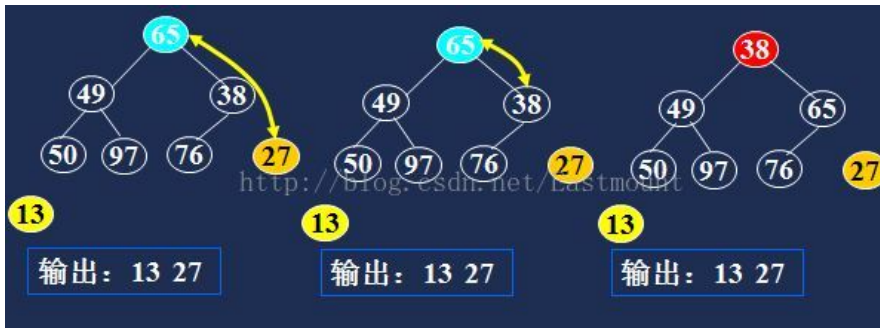
(1).输出堆顶元素,题中堆顶元素=13;输出后以堆中最后一个元素=97替代堆顶元素;然后将根节点值与左右子树的根节点进行比较,并与其中小者进行交换;



(2).重复上述操作,直到叶子节点,将得到的新堆称为这个从堆顶至叶子的调整过程为"筛选".

输出堆顶27,将最后一个元素65替代堆顶元素,再从49和38中选择较小值38,将65与38元素交换.(答题是只画最后交换好的再配上必要文字)

说明即可)



输出堆顶38,将最后一个元素76替代堆顶元素,再根据方法依次交换76与49,76与50.



输出堆顶49,将最后一个元素97替代堆顶元素,再根据方法依次交换97与50,97与76.



同理一次输出如下:



三.总结

该文章主要是解决上面的那个堆排序问题,而相应的算法代码和插入删除亦类似没给出.写这篇文章主要是纪念自己这几个月考研的日子,同时无论考研结果如何,都要在考完后学习更多自己感兴趣的编程知识,并完成一些自己感兴趣的工程和博客.这才是我想要的东西.最后希望该文章对大家有所帮助,尤其是考研的学子.希望大家考研顺利,同时如果文章中有错误或不足之处,希望大家海涵!

(BY:Eastmount 2013-12-17 下午4点 <http://blog.csdn.net/eastmount/>)



Eastmount



博客专家

原创文章 462

获赞 6725

访问量 525万+

关注

他的留言板