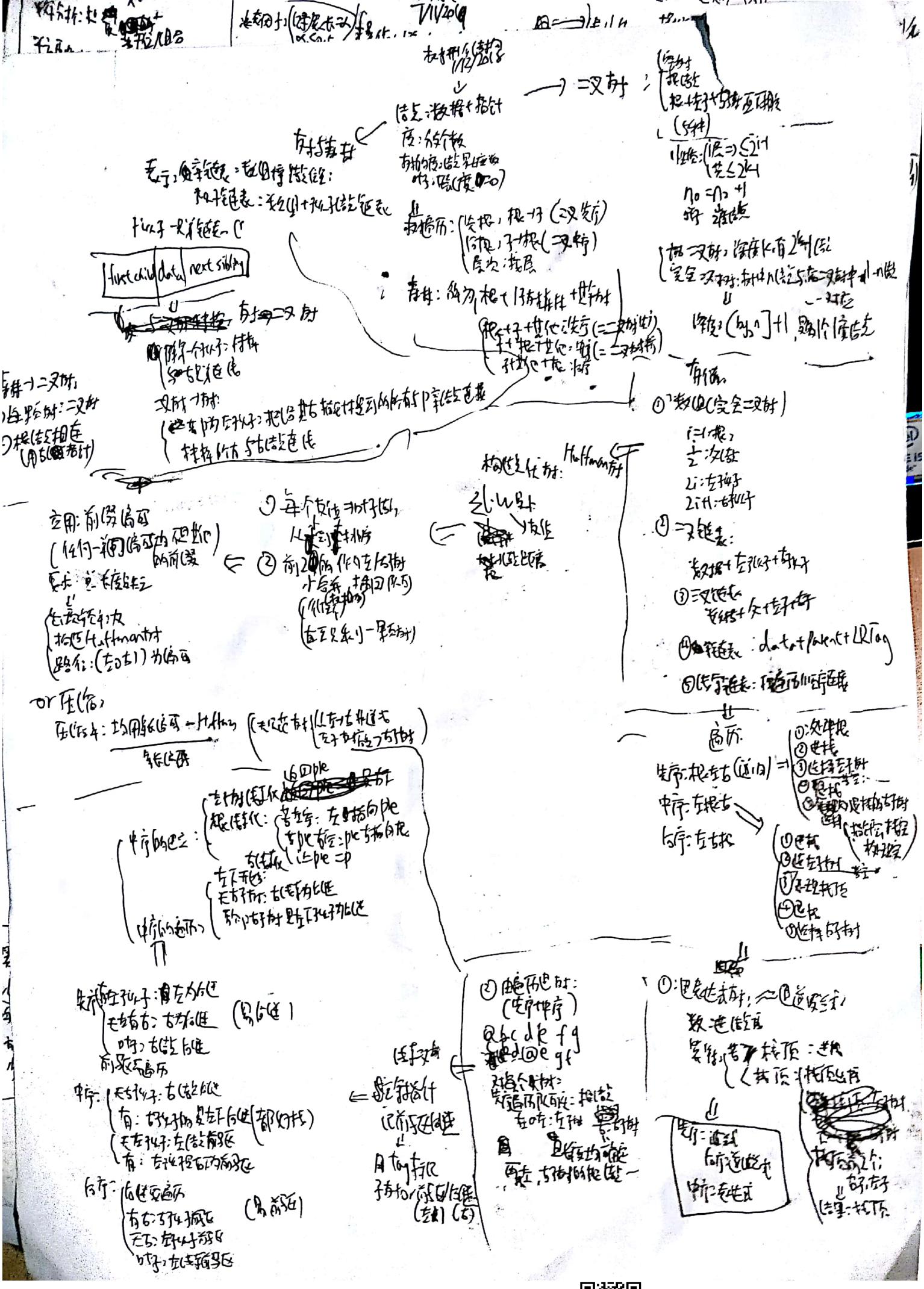
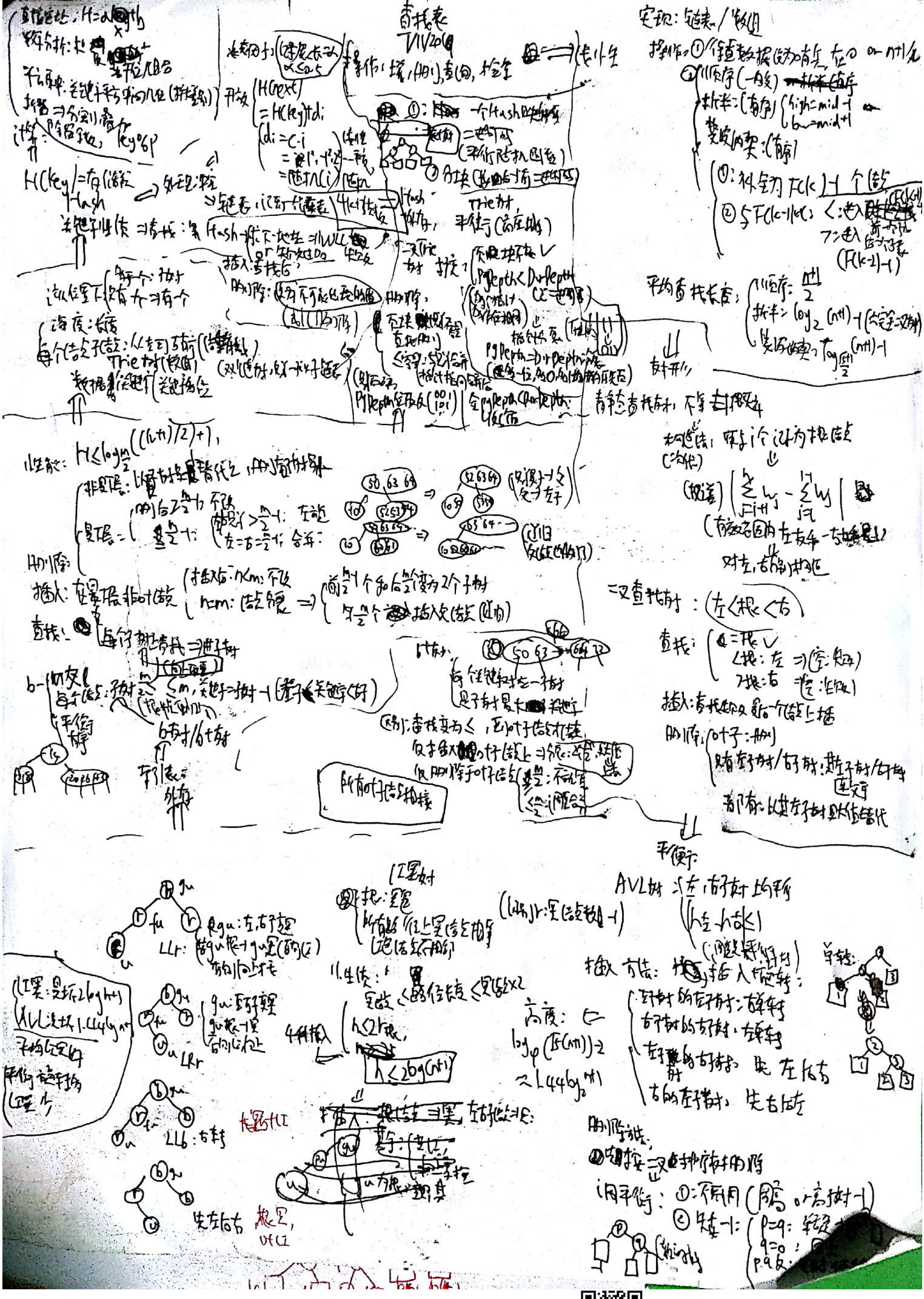
learn and let learn

qsir.org{qiutopia}



由 扫描全能王 扫描创建



1/7/2019

排序

名称	思路	方法 　以己为师 自求自得	内存	是否稳定	复杂度
插入	顺序部分依次插入有序部分 查找 + 移动	① 设置哨兵：R[0]=R[i] ② 查找：顺序查找 / 二分查找 ③ 插入位置后移 ④ 插入：顺序空间：后移；链表：一次性调整	直接:1 折半:1	均稳定	比较：直接：$\frac{n^2}{4}$；折半：$nlog_2n$；移动：全：$\frac{n^2}{4}$；$O(n^2)$
希尔	先宏观（跳跃插入）再微观	e.g. ① d=5：分为5个子序列排 ② d=3：3个…… ③ d=1：	1	X	$=n^{1.3}$
冒泡	把最大/小冒到最上	一趟：依次两两比较 i++ 最后一次无交换停止	1	V	n^2；最好：n-1 次比较 0 移动；最坏：比较 $\frac{n(n-1)}{2}$，移动 $\frac{3n(n-1)}{2}$
快排	① 取枢轴 ② 把<：左 >：右 ③ 排左右2部分	一趟：high与轴比：轴大<：换到 low 移动 high low与轴比：轴小>：high 移动 low 最后 high=low：移动轴，左<轴 右>轴 还有递归栈	log_2n ~ n（栈）	X	一趟：n；最好：log_2n+1 次；最差：n 次；平均：$O(nlog_2n)$
选排	一次选出最小值插到有序尾	① 简单（5种方法）：选择方法：① 初始至2个（简单的比较）② 依次比较，选出最小值（交换：冯向依次比较） ② 树形排序 改进方法：删除选择	简:0	X	简单比较 $\frac{n(n-1)}{2}$，移动 $3(n-1)$；$O(n^2)$（交换X3）
		锦标赛：树形比较排序 → 根的 → 一条路径 ① 构建锦标赛：n→比赛树 → 补满至 2^k ② 比较：胜者：下次比；败者：淘汰 改进方法：删除： 依次与兄弟比较：胜：下次；败：淘汰	胜败树 2n-1 O(n)	均稳定	胜败树：没有剩余；第一次：n-1 次比较；之后：$(n-1)\cdot log_2n$；共 $nlog_2n$ 次（5种输入无关）
堆排	利用完全二叉树特性 小堆：$r_i \le r_{2i}, r_i \le r_{2i+1}$ 大顶	① 建堆：从下往上筛选 ② 排序：堆顶和末尾对换，堆底→顶 筛选：对某一层：先比左右子树，指向更大的根；若根大于左右子：停止 否则：交换根与大的，比下一层（原来的大根）	1	X	建立：4n；筛选：一共 2(h-1) ($2\cdot log_2n$)；总共 $2nlog_2n+4n$；$O(nlog_2n)$（5种数与输入无关）
归并	2路归并；递归执行	从 len=1 开始：相邻2个长为 len 的归并，直至剩余不足 2len：≤ len：复制；> len：再归并 len×=2，再归并	n	V	每一趟：n/2len · 2len=n；共 log_2n 趟；共 $nlog_2n$
基数	把关键字（按位/按字）转化为多关键字	① 分配：按某一位分配到队列（如数字：0~9 10个队列） （从低位到高位依次分配） ② 收集：按某一位从前向后收集（顺序）	n+r	V	$O(d(n+r))$ d：关键字长度；r：基数范围

清華大學

$2(n_0-1)=2(k+(m+1)n_0-k)$

$2n_0 \ldots (m+1)n_0-mk-2n_0+2$

$\Rightarrow n_0=kn_0-2$

外部归并排序:

内部初始排序 + 内部归并排序 + 内外存信息交换

⇓

① n路归并时, 从n路中选一个最小记录

⇓

败者树

↓

内点: 指向败者位置的指针

② 置换-选择排序

选择最小记录与输入输出同步

① 填充工作区

② 工作区: 关键字值最小 → minimax, 输出

③ 输入下一个到工作区

④ 从比minimax大的中选一个最小的, 记为新minimax, 输出 } 一段

③ 最佳归并树: 作对不同长度的归并段, 每个归并段归并读写次数 次

⇓

用Huffman树优化 (类似于哈夫曼树, 求m叉Huffman树)

⇓

若不足时)

必须使 $\frac{n-1}{m-1}$ 为整数

补充虚段(长为0) 使之成立

⇓

Huffman树

1/13/2019

1. 数组每个元素4字节，起始位置100，以行为优先，A 4×3×5×8e.

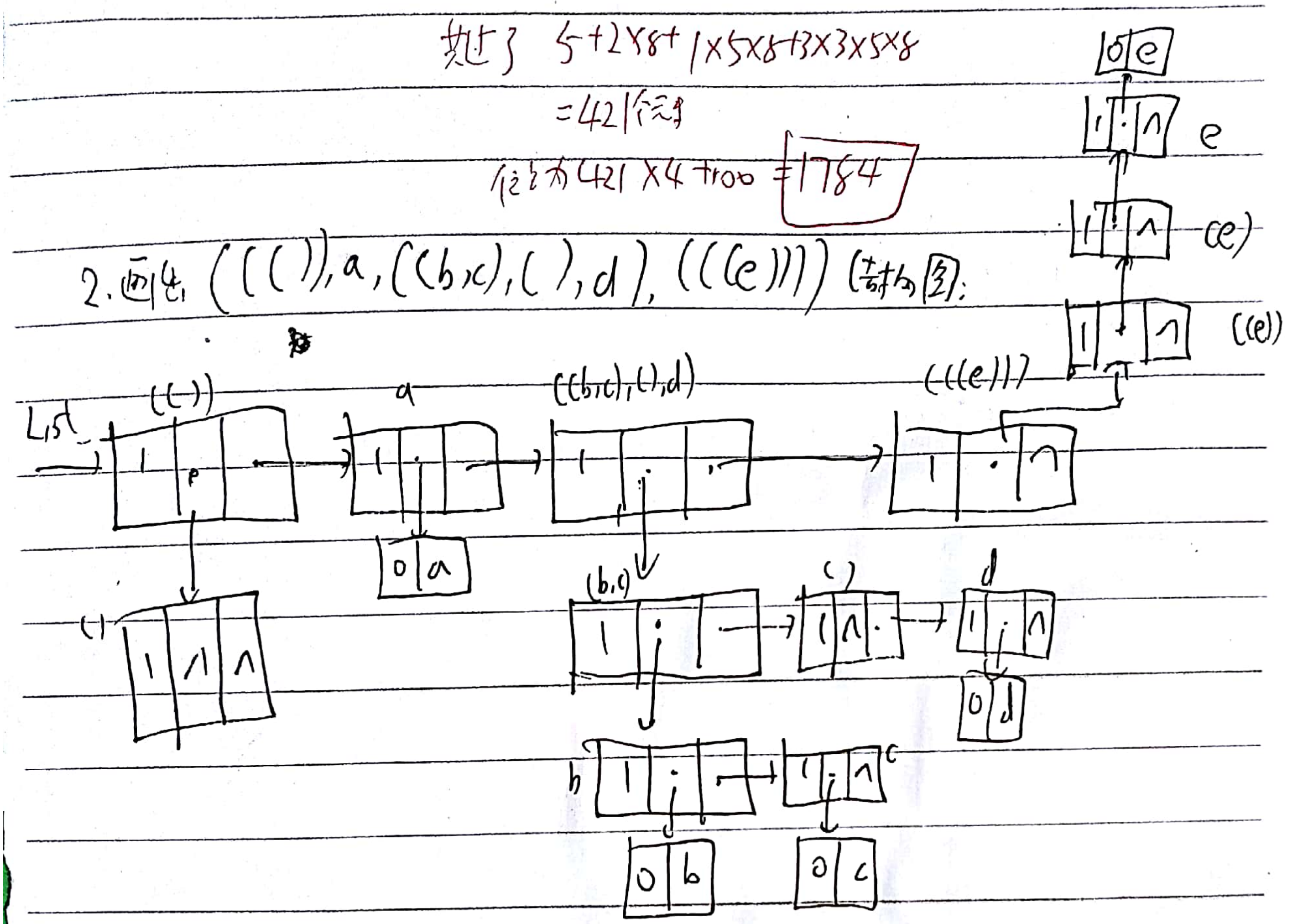
求 a_{3125} 位置　　以行为优先：x>y>z>w（优先）

跳过了 5+2×8+1×5×8+3×3×5×8

=421个元素

位置为 421×4+100 = 1784

2. 画出 ((()),a,((b,c),(),d),(((e)))) 存储结构图：

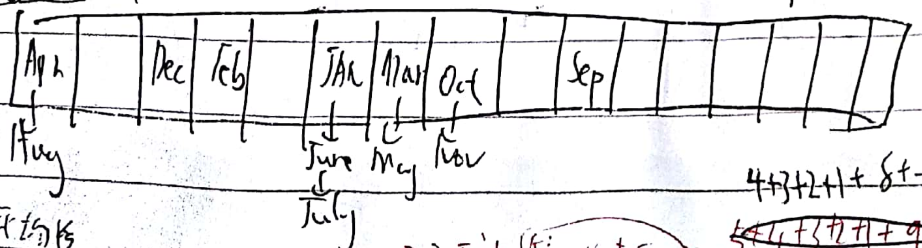


3. 设 $H(x)=\lfloor \frac{i}{2} \rfloor$ （i：首字母在字母表序列）

① 开放定址 $d_i=i$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Apr	Aug	Dec	Feb		JAN	Mar	May	June	July	Sep	Oct	Nov				

② 链地址



求查找成功的平均长度

4+3+2+1+1+8+…+1

① 开放定址：$\frac{\ldots}{14} = \frac{46}{14}$

② 链地址：$\frac{2+1+1+3+2+2+1}{14} = \frac{12}{14}$

①所有可能 H(x) 从0到13, 14种

②查找不成功的（从表末…）

4. 假设一次I/O的开销时间为1s，每次可由内部排序750记录，则对150000记录进行4路平衡归并需几次I/O

答：① 一个归并段：750

共200个归并段

② 内部排序：每次读1块写一块；

共1000块，2000次I/O

③ 外部归并 $\log_4 200 = 4$ ∴ 4趟

每趟：每个记录读1写1

共1000块读1写1

I/O $1000\times2+(1000\times2)\times4=10000$ 次I/O

1. 程序 CreateLosertree 是创建初始败者树的算法：

```
Void CreateLosertree(LoserTree &ls){
    b(k).key = minkey;
    for ((i=0; i<k; ++)  ls[ i ]=k;
    for ( i=k-1; i>=0; i-- )  Adjust ( ls, i);
}//CreateLosertree

Void Adjust ( LoserTree &ls,  int q){
    int t = (q+k)/2
    while( t>0 ){
    if( b[q].key > b[ls[t]].key )  Swap(q, ls[t]);
    t=t/2;
    }//while
    ls[0]=q
}// Adjust
```

存储败者 ls[1]~ls[k-1] 存的是败者的在数组下标

ls[0] 冠军

而 (q+k)/2 相当于找双亲

ls[0]=冠军值