## 第五章

1、（1）判定树



（2）查找49，依次比较：38,53,40,49

（3）查找80，依次比较：38,53,79,85

（4）每个元素的查找长度为该结点在判定树中的层数，如40的查找长度为3，所以，查找成功平均查找长度：（3+4+2+3+4+1+3+4+2+4+3+4）/ 12 = 3.08

2、分析：最理想的分块方式应该是使得平均查找长度最小。根据p206公式：

这里n=4000，求m（块内记录数）的值（）使得ASL取得最小值。

求导可得（或者编写计算机程序求得结果）当m=5时，ASL最小，为7.15，即分800块，每块长度为5时最理想。

若每块长度为50，带入上述公式，得出平均查找长度为27.12

3、

B-树：



删除29之后：



删除47之后：



4、简述B-树与B+树的区别。

（1）同阶B+树结点最大关键字数比B-树多1

（2）B+树所有关键字均在叶子结点，而B-树关键字不止存在叶子结点

（3）B+树叶子结点相互连接为双向链表，而B-树叶结点之间没有直接联系

（4）B+树非叶结点的指针p[i]指向关键字区间在[k[i], k[i+1])之间的子树，而B-树为全开区间

5、（1）线性探查法：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | 4 |  | 12 | 49 | 38 | 13 | 24 | 32 | 21 |  |

hash(32) = 8

hash(13) = 6

hash(49) = 4

hash(24) = 6 冲突，选择7

hash(38) = 4 冲突，选择5

hash(21) = 8 冲突，选择9

hash(4) = 1

hash(12) = 3

查找成功平均长度：(1+1+1+2+2+2+1+1)/8 = 11/8

查找失败平均长度：(3+2+8+7+6+5+4+3+2)/11 = 40/11

（2）链接法：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1  4 | 2 | 3  12 | 4  38  49 | 5 | 6  24  13 | 7 | 8  21  32 | 9 | 10 |

查找成功平均长度：(1+1+1+2+1+2+1+2)/8 = 11/8

查找失败平均长度：(6+1+1+2+2+2)/11 = 14/11

6、

插入关键码算法：

设散列表大小为s，关键码得到的hash地址为x

从x位置开始依次查看(x+i)%s (i=0,1,…,s-1)位置，若当前位置不包含有效关键码（为空或者已删除），则将关键码存入，插入成功，返回；否则，插入失败，返回。

删除关键码算法：

设散列表大小为s，关键码为k，关键码得到的hash地址为x

从x位置开始依次查看(x+i)%s (i=0,1,…,s-1)位置，若当前位置包含有效关键码且关键码为k，删除关键码并将当前位置标记为已删除；否则删除失败，返回。

7、（1）线性探查法：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 89 | 71 |  | 23 | 73 | 44 |  |  |  | 99 |

hash(71) = 1

hash(23) = 3

hash(73) = 3 冲突，选择4

hash(99) = 9

hash(44) = 4 冲突，选择5

hash(89) = 9 冲突，选择0

查找成功平均查找长度：(1+1+2+1+2+2)/6 = 3/2

查找失败平均查找长度：(4+3+2+4+3+2+4)/10 = 11/5

（2）双散列：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  | 71 |  | 23 | 44 |  |  | 73 |  | 99 |

hash(71) = 1

hash(23) = 3

hash(73) = 3冲突

rehash(73) = 4

H1 = (hash(73) + 1 \* rehash(73)) % 10 = 7

hash(99) = 9

hash(44) = 4

hash(89) = 9冲突

rehash(89) = 5

H1 = (hash(89) + 1 \* rehash(89)) % 10 = 4冲突

H2 = (hash(89) + 2 \* rehahs(89)) % 10 = 9 冲突，且回到hash(89)位置，插入失败

查找成功平均查找长度：(1+1+2+1+1) / 5 = 6/5

查找失败平均查找长度：（由于rehash的值与关键码有关，因此很难确定失败的查找长度）

（3）链接法：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1  71 | 2 | 3  23  73 | 4  44 | 5 | 6 | 7 | 8 | 9  89  99 |

查找成功平均查找长度：(1+1+2+1+1+2)/6 = 4/3

查找失败平均查找长度：(6+1+2+1+2)/10 = 6/5

8、

线性探查再散列的查找成功的平均查找长度的理论值为(1+ 1/(1-a))/2，因为比较次数小于3，得到装填因子a <= 4/5，为0.8，因此存储空间约为250/0.8 = 312.5，上取整为313。

## 第六章

1、

（1）直接插入：

60

40, 60

40, 60, 120,

40, 60, 120, 185

20, 40, 60, 120, 185,

20, 40, 60, 120, 135, 185

20, 40, 60, 120, 135, 150, 185

20, 40, 60, 120, 130, 135, 150, 185

20, 40, 45, 60, 120, 130, 135, 150, 185

（2）冒泡

60, 40, 120, 185, 20, 135, 150, 130, 45

40, 60, 120, 20, 135, 150, 130, 45, 185

40, 60, 20, 120, 135, 130, 45, 150, 185

40, 20, 60, 120, 130, 45, 135, 150, 185

20, 40, 60, 120, 45, 130, 135, 150, 185

20, 40, 60, 45, 120, 130, 135, 150, 185

20, 40, 45, 60, 120, 130, 135, 150, 185

（3）快排

60, 40, 120, 185, 20, 135, 150, 130, 45

45, 40, 20, 60, 185, 135, 150, 130, 120

20, 40, 45, 60, 120, 135, 150, 130, 185

20, 40, 45, 60, 120, 135, 150, 130, 185

20, 40, 45, 60, 120, 130, 135, 150, 185

（4）堆排序

60, 40, 120, 185, 20, 135, 150, 130, 45

建立最大堆：

185, 130, 150, 120, 20, 60, 135, 40, 45

堆排序：

150, 130, 135, 120, 20, 60, 45, 40, 185

135, 130, 60, 120, 20, 40, 45, 150, 185

130, 120, 60, 45, 20, 40, 135, 150, 185

120, 45, 60, 40, 20, 130, 135, 150, 185

60, 45, 20, 40, 120, 130, 135, 150, 185

45, 40, 20, 60, 120, 130, 135, 150, 185

40, 20, 45, 60, 120, 130, 135, 150, 185

20, 40, 45, 60, 120, 130, 135, 150, 185

（5）归并（递归结果）

60, 40, 120, 185, 20, 135, 150, 130, 45

（60, 40, 120, 185）（20, 135, 150, 130, 45）

（60, 40）（120, 185）（20, 135, 150, 130, 45）

（60）（40）（120）（185）（20, 135, 150, 130, 45）

（40, 60）（120, 185）（20, 135, 150, 130, 45）

（40, 60, 120, 185）（20, 135, 150, 130, 45）

（40, 60, 120, 185）（20, 135）（150, 130, 45）

（40, 60, 120, 185）（20）（135）（150, 130, 45）

（40, 60, 120, 185）（20, 135）（150, 130, 45）

（40, 60, 120, 185）（20, 135）（150）（130, 45）

（40, 60, 120, 185）（20, 135）（150）（130）（45）

（40, 60, 120, 185）（20, 135）（150）（45, 130）

（40, 60, 120, 185）（20, 135）（45, 130, 150）

（40, 60, 120, 185）（20, 45, 130, 135, 150）

20, 40, 45, 60, 120, 130, 135, 150, 185

2、

增量序列改成：{7, 3, 1}

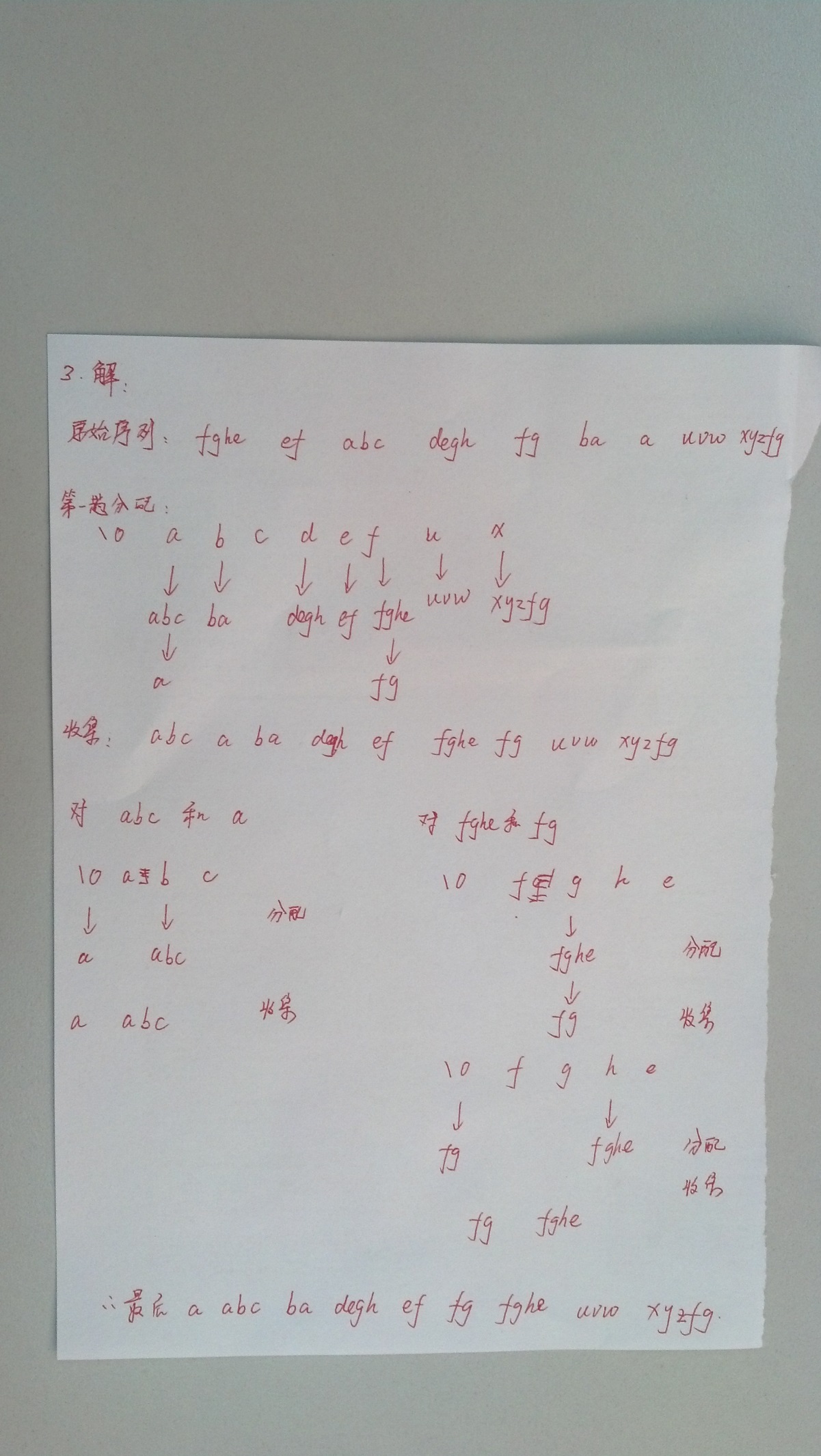
9 8 6 5 4 3 2 1 0

1 8 6 5 4 3 2 9 0 增量为{7}

1 4 0 2 8 3 5 9 6 增量为{3}

0 1 2 3 4 5 6 8 9 增量为{1}

3、



5、

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 比较次数 | | | 移动次数 | | |
|  | 相等 | 正序 | 逆序 | 相等 | 正序 | 逆序 |
| 直接插入排序 |  |  |  |  | 0 |  |
| 希尔排序 |  |  |  | 0 | 0 |  |
| 快速排序 |  |  |  |  |  |  |
| 归并排序 |  |  |  |  |  |  |

8、

5个数据时，基于比较的排序算法至少要比较多少次。

最少比较次数：4次。

7个数据时，基于比较的排序算法至少要比较多少次。

最少比较次数：6次