第4章

4.3 StrLength (s) = 14

StrLength (t) = 4

SubString (s, 8, 7) = "STUDENT" (字符在串中的位置从1开始编号)

SubString (t, 2, 1) = "O"

Index (s, 'A') = 3

Index (s, t) = 0

Replace (s, 'STUDENT', q) = "I AM A WORKER"

Concat (SubString (s, 6, 2), Concat (t, SubString (s, 7, 8)))

= "A GOOD STUDENT"

4.4 s = "THIS SAMPLE IS "

t = "A GOOD "

u = "ODNE" , v = "THIS SAMPLE IS A GOOD ODNE"

StrLength (s) = 14

Index (v, g) = 13

Index (u, g) = 0

A D A B B A D A D A

4.8 nextval = [-1, 0, -1, 0, 0, -1, 0, 1, 2, 1]

第1趟 ADBADABBAABADABBADADA

ADABBADADA j=2 nextval[j]=-1

第2趟 ADBADABBAABADABBADADA

ADABBADADA j=6 , nextval[j]=0

第3趟 ADBADABBAABA-DABBADADA

ADABBADADA j=1 nextval[j]=0

第4趟 ADBADABB.AAB.ADABBADADA

ADA ... j=0 nextval[j]=-1

第5趟: ADBADABBAABADABBADADA

ADABBADADA 成功,返回11

5.1

（1）6×8×6=288 bytes

（2）1000+(5×8+7)×6=1282

（3）1000+(1×8+4)×6=1072

（4）1000+(7×6+4)×6=1276

5.8

i 为奇数，k=i+j-2

i 为偶数，k=i+j-1

5.11

（1）GetHead【GetTail【【GetTail【L1】】】】

（2）GetHead【GetHead【【GetTail【L2】】】】

（3）GetHead【GetHead【【GetTail【GetTail【【GetHead【L3】】】】】】】

（4）GetHead【GetHead【【GetHead【GetTail【【GetTail【L4】】】】】】】

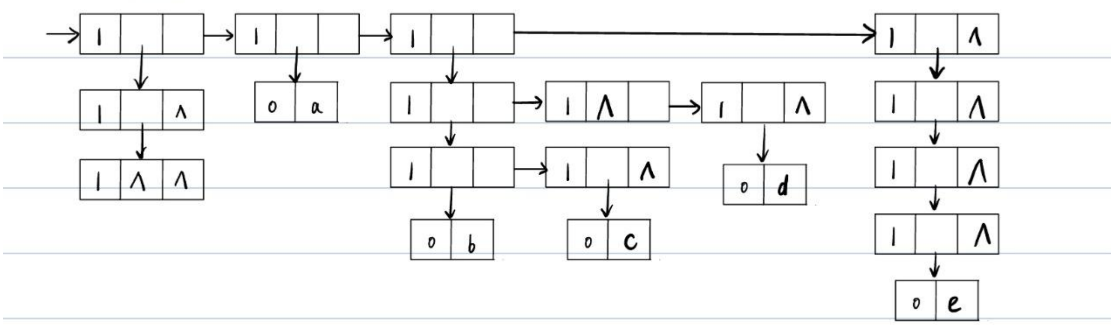
（5）GetHead【GetHead【GetTail【【GetTail【L5】】】】】

（6）GetHead【GetTail【【GetHead【L6】】】】

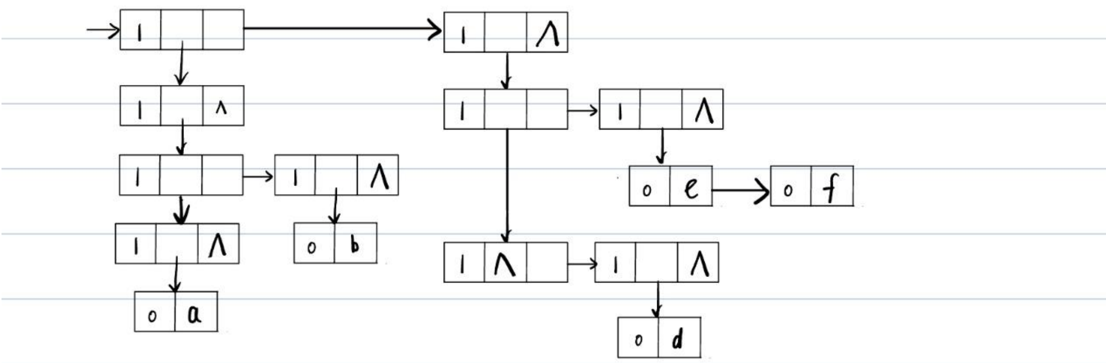
（7）GetHead【GetHead【【GetTail【GetHead【【GetTail【L7】】】】】】】

5.12

(1) 深度：4



(2) 深度：4



5.15

当 S 为空集时，幂集仅包含空集：$P(\varnothing)=\{\varnothing\}$

若 S 非空，取出一个元素 $x\in S$，剩余集合为 $S'=S\setminus\{x\}$。则幂集由包含 x 的子集和不包含 x 的子集组成：$P(S)=P(S')\cup\{\{x\}\cup A \mid A\in P(S')\}$

6.2

二叉树是度为 2 的树，其子树有左右之分且次序不能颠倒。而度为 2 的树不一定有序。

6.5

除根节点外，每个结点有且仅有一个前驱节点，即与一条边对应，故 总结点数=总度数+1

则 叶节点数 = 度为 0 结点数 = 总度数+1-度不为 0 结点数

$$=1+n_1+2n_2+\dots+kn_k-(n_1+n_2+\dots+n_k)$$

$$=1+\sum_{i=1}^{k}(i-1)n_i$$

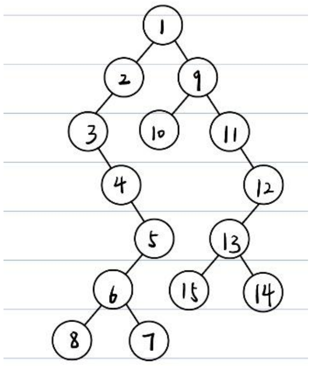
6.18

若*p 是根节点，则没有后继；

若*p 是其双亲结点的右孩子，则后继是双亲结点；

若*p 是其双亲结点的左孩子，则后继是其兄弟结点最左下的子孙。

6.20



(1) 1 2 3 4 5 6 8 7 9 10 11 12 13 15 14

(2) 3 4 8 6 7 5 2 1 10 9 11 15 14 13 12

(3) 8 7 6 5 4 3 2 10 15 14 13 12 11 9 1

6.31

证明：由一棵二叉树的先序序列和中序序列可以唯一确定这棵二叉树

归纳证明：设二叉树的深度为 k

k=0 时，空树，显然。

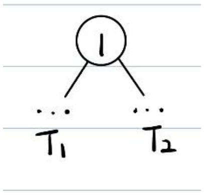
k=1 时，只有根节点，显然。

k=2 时，对于三种可能的结构均显然。

假设 k≤n 时结论成立

k=n+1 时，

设树 T 的结构如图，设根节点编号为 1，T1、T2 分别表示根节点的左子树、右子树



用 Pre(T)表示树 T 的先序序列，Mid(T)表示树 T 的中序序列

则 Pre(T)=1 Pre(T1) Pre(T2) , Mid(T)=Mid(T1) 1 Mid(T2)

根据归纳假设，Pre(T1)和 Mid(T1)、Pre(T2)和 Mid(T2)可以分别唯一确定深度≤n 的树 T1、T2 的结构，那么 T 的结构也就被唯一确定了。