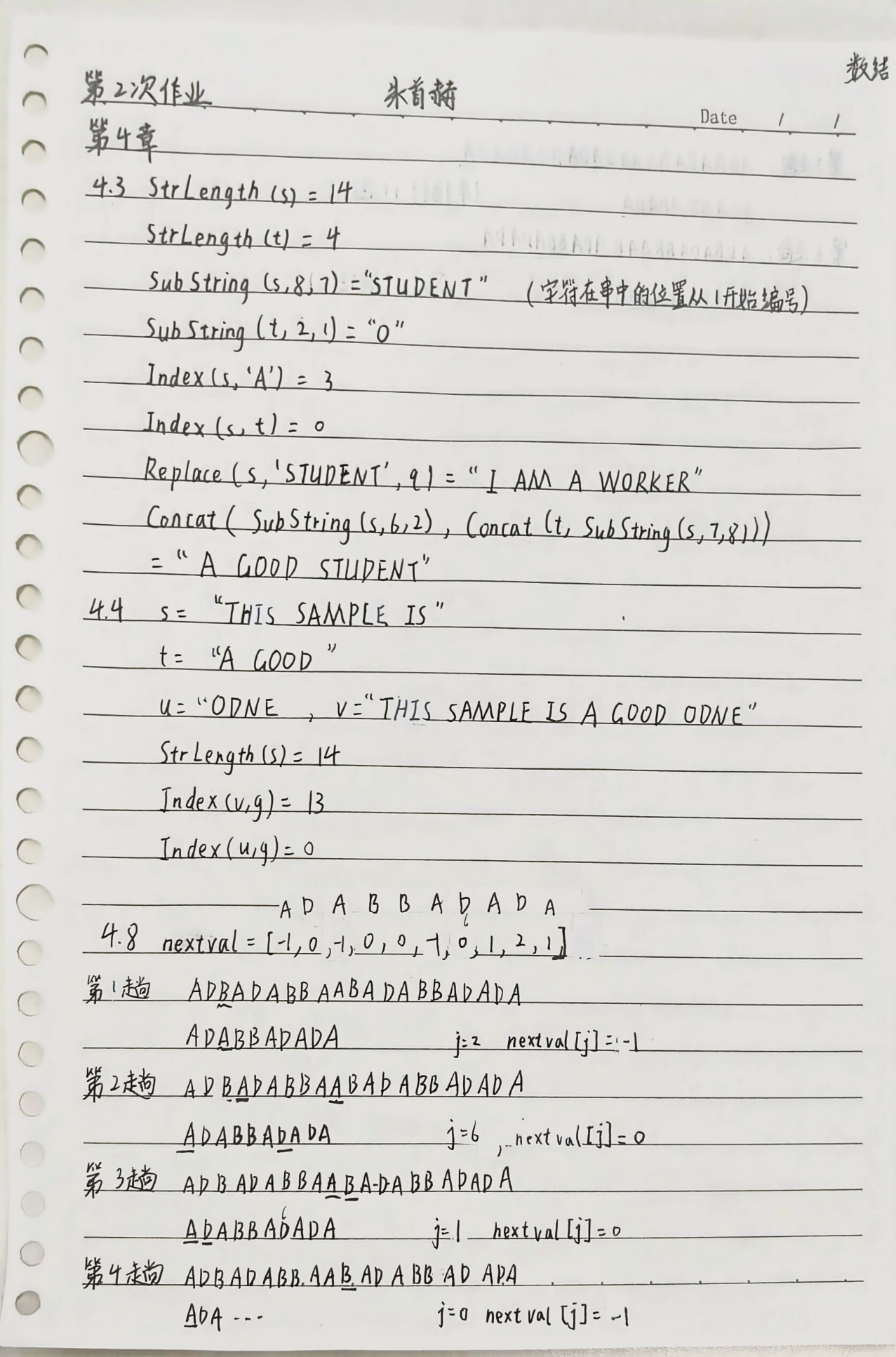
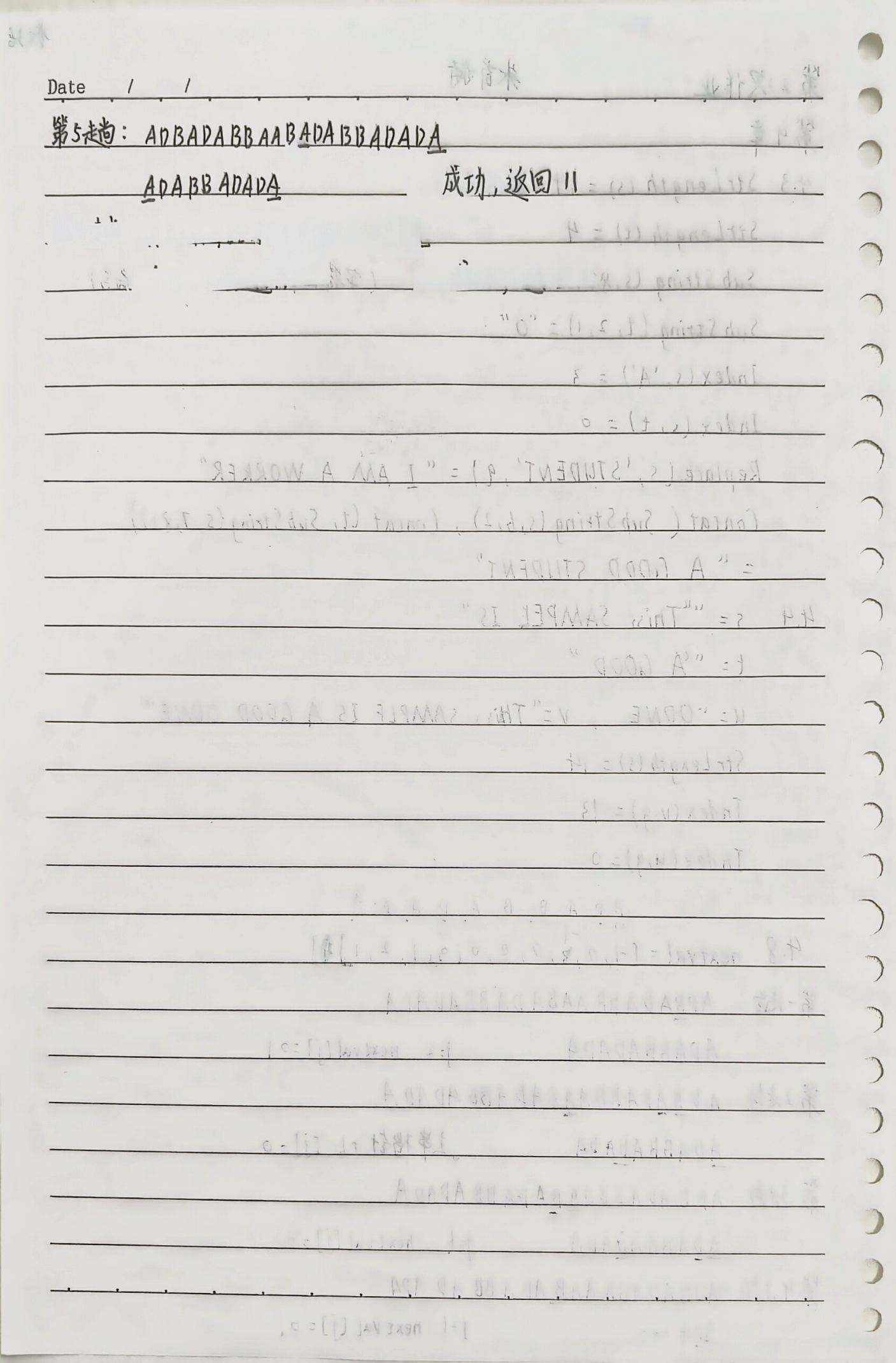
****

****

**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】】】】】】】

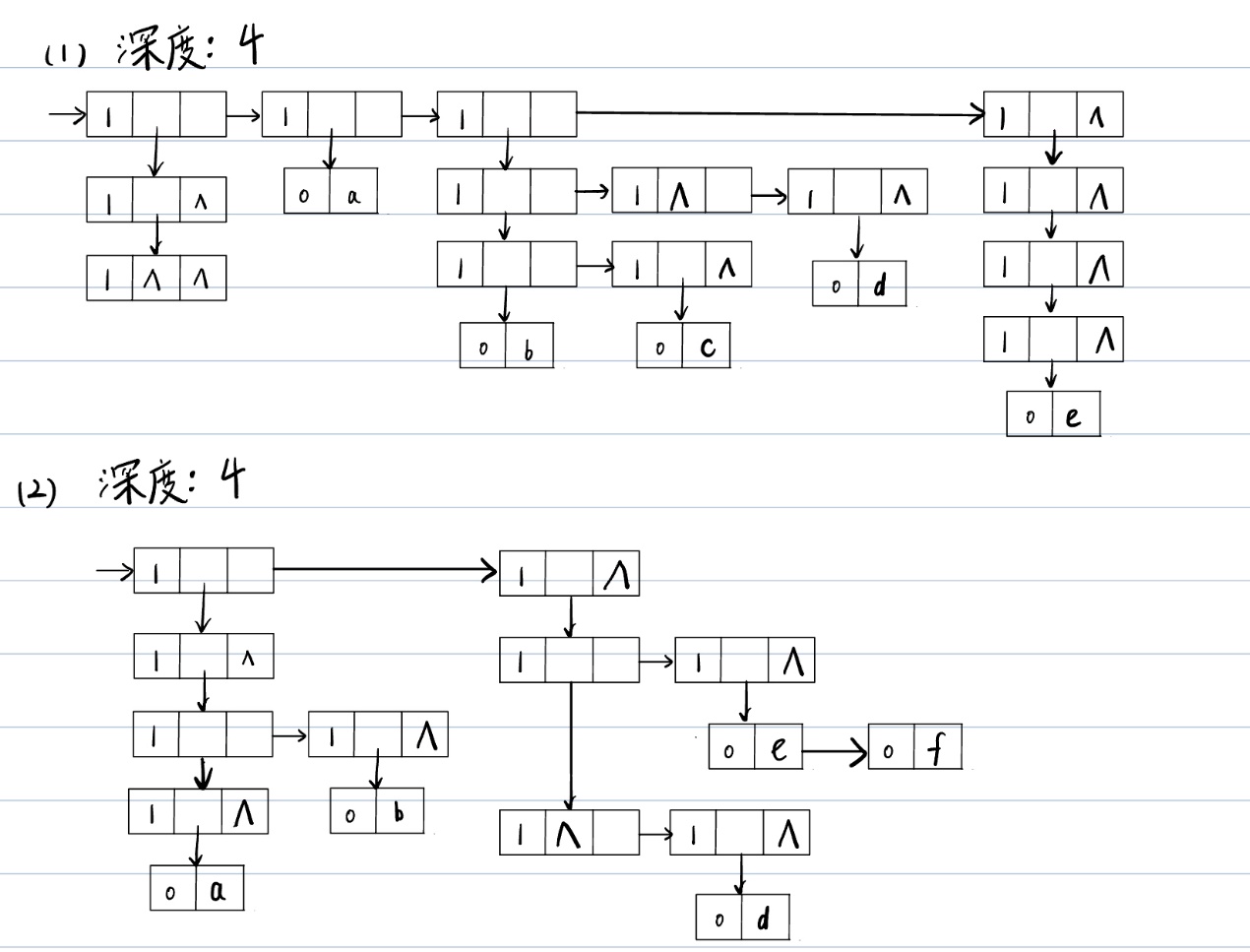
（4）GetHead【GetHead【【GetHead【GetTail【【GetTail【L4】】】】】】】

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

（6）GetHead【GetTail【【GetHead【L6】】】】

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

**5.12**

****

**5.15**

当 S 为空集时，幂集仅包含空集：P(∅)={∅}

若 S 非空，取出一个元素 x∈S，剩余集合为 S′=S\{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′)∪{{x}∪A∣A∈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+n1+2n2+…+knk – (n1+n2+…+nk)

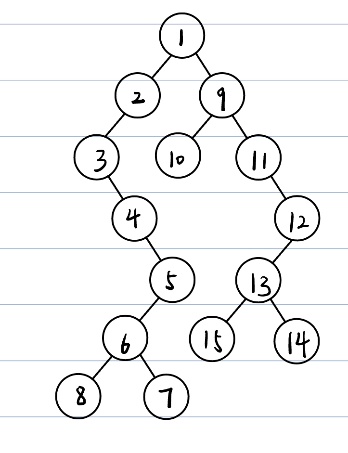
=1+

**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。

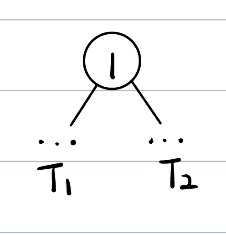
k=1时，只有根节点，显然。

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。