

第1章习题

✧ 1.8 设 n 为正整数，试确定下列各程序段中前置以记号 @ 的语句的频度。

- (1) $i=1; k=0;$
while($i \leq n-1$){
@ $k+=10*i;$
 $i++;$ }
- (2) $i=1; k=0;$
do{ @ $k+=10*i;$
 $i++;$ } while($i \leq n-1$);
- (3) $i=1; k=0;$
while($i \leq n-1$){
 $i++;$
@ $k+=10*i;$ }
- (4) $k=0;$
for($i=1; i \leq n; i++$){
for($j=i; j \leq n; j++$)
@ $k++;$ }
- (5) for($i=1; i \leq n; i++$){
for($j=1; j \leq i; j++$){
for($k=1; k \leq j; k++$)
@ $x+=\text{delta};$ }
- (6) $i=1; j=0;$
while($i+j \leq n$){
@ if($i>j$) $j++;$
else $i++;$ }
- (7) $x=n; y=0;$ // n 是不小于 1 的常数
while($x>=(y+1)*(y+1)$){
@ $y++;$ }
- (8) $x=91; y=100;$
while($y>0$){
@ if($x>100$) { $x=100; y--;$ }
else $x++;$ }

1

✧ 1.9 假设 n 为 2 的乘幂，并且 $n > 2$ ，试求下列算法的时间复杂度及变量 count 的值（以 n 的函数形式表示）。

```
int Time ( int n ) {
    count = 0; x = 2;
    while ( x < n/2 ) {
        x *= 2; count++;
    }
    return ( count );
} // Time
```

2

✧ 1.12 设有以下三个函数：

$$f(n) = 21n^4 + n^2 + 1000 \quad g(n) = 15n^4 + 500n^3 \quad h(n) = 5000n^{3.5} + n \log n$$

请判断以下断言正确与否：

- (1) $f(n)$ 是 $O(g(n))$
- (2) $h(n)$ 是 $O(f(n))$
- (3) $g(n)$ 是 $O(h(n))$
- (4) $h(n)$ 是 $O(n^{3.5})$
- (5) $h(n)$ 是 $O(n \log n)$

✧ 1.20 试编写算法求一元多项式 $P_n(x) = \sum_{i=0}^n a_i x^i$ 的值

$P_n(x_0)$ ，并确定算法中每一语句的执行次数和整个算法的时间复杂度。注意选择你认为较好的输入和输出方法。本题的输入为 $a_i (i=0, 1, \dots, n)$ ， x_0 和 n ，输出为 $P_n(x_0)$ 。

3

第2章习题

✧ 2.6 已知 L 是无表头结点的单链表，且 P 结点既不是首元结点，也不是尾元结点，试从下列提供的答案中选择合适的语句序列。

- a. 在 P 结点后插入 S 结点的语句序列是_____。
- b. 在 P 结点前插入 S 结点的语句序列是_____。
- c. 在表首插入 S 结点的语句序列是_____。
- d. 在表尾插入 S 结点的语句序列是_____。

4

- (1) $P \rightarrow \text{next} = S;$
- (2) $P \rightarrow \text{next} = P \rightarrow \text{next} \rightarrow \text{next};$
- (3) $P \rightarrow \text{next} = S \rightarrow \text{next};$
- (4) $S \rightarrow \text{next} = P \rightarrow \text{next};$
- (5) $S \rightarrow \text{next} = L;$
- (6) $S \rightarrow \text{next} = \text{NULL};$
- (7) $Q = P;$
- (8) while($P \rightarrow \text{next} \neq Q$) $P = P \rightarrow \text{next};$
- (9) while($P \rightarrow \text{next} \neq \text{NULL}$) $P = P \rightarrow \text{next};$
- (10) $P = Q;$
- (11) $P = L;$
- (12) $L = S;$
- (13) $L = P;$

5

✧ 2.7 已知 L 是带头结点的非空单链表，且 P 结点既不是首元结点，也不是尾元结点，试从下列提供的答案中选择合适的语句序列。

- a. 删除 P 结点的直接后续结点的语句序列是_____。
- b. 删除 P 结点的直接前驱结点的语句序列是_____。
- c. 删除 P 结点的语句序列是_____。
- d. 删除首元结点的语句序列是_____。
- e. 删除尾元结点的语句序列是_____。

6

- (1) P=P->next;
- (2) P->next=P;
- (3) P->next=P->next->next;
- (4) P=P->next->next;
- (5) while(P!=NULL) P=P->next;
- (6) while(Q->next!=NULL) { P=Q; Q=Q->next; }
- (7) while(P->next!=Q) P=P->next;
- (8) while(P->next->next!=Q) P=P->next;
- (9) while(P->next->next!=NULL) P=P->next;
- (10) Q=P;
- (11) Q=P->next;
- (12) P=L;
- (13) L=L->next;
- (14) free(Q);

7

✧ 2.8 已知P结点是某双向链表的中间结点，试从下列提供的答案中选择合适的语句序列。

- a. 在P结点后插入S结点的语句序列是_____。
- b. 在P结点前插入S结点的语句序列是_____。
- c. 删除P结点的直接后续结点的语句序列是_____。
- d. 删除P结点的直接前驱的语句序列是_____。
- e. 删除P结点的语句序列是_____。

8

- (1) P->next=P->next->next;
- (2) P->priou=P->priou->priou;
- (3) P->next=S;
- (4) P->priou=S;
- (5) S->next=P;
- (6) S->priou=P;
- (7) S->next=P->next;
- (8) S->priou=P->priou;
- (9) P->priou->next=P->next;
- (10) P->priou->next=P; (11) P->next->priou=P;
- (12) P->next->priou=S; (13) P->priou->next=S;
- (14) P->next->priou=P->priou;
- (15) Q=P->next; (16) Q=P->priou;
- (17) free(P); (18) free(Q);

9

✧ 2.17 试写一算法，在无头结点的动态单链表上实现线性表操作INSERT(L, i, b)，并和在带头结点的动态单链表上实现相同操作的算法进行比较

✧ 2.11 设顺序表va中的数据元素递增有序。试写一个算法，将x插入到顺序表的适当位置上，以保持该表的有序性。

10

✧ 2.19 已知线性表中的元素以值递增有序排列，并以单链表作存储结构。试写一高效的算法，删除表中所有值大于mink且小于maxk的元素(若表中存在这样的元素)，同时释放被删除结点空间，并分析你的算法的时间复杂度(注意：mink和maxk是给定的两个参变量，它们的值可以和表中的元素相同，也可以不同)

✧ 附加题.在顺序存储结构的线性表a基础上重做上题。

11

✧ 2.10 指出下列算法中的错误和低效（即费时）之处，并将它改为一个即正确又高效的算法。

```
status DeleteK(SqList &a, int i, int k){
//本过程从顺序存储结构的线性表a中删除第i个元素起的k个元素
if(i<1||k<0||i+k>a.length) return INFESIBLE; //参数不合法
else{
    for(count=1; count<k; count++){//删除一个元素
        for(j=a.length; j>=i+1; j--) a.elem[j-1]=a.elem[j]
        a.length--;
    }
    return OK;
}
} //DeleteK
```

12

✧ 2.22 试写一算法，对单链表实现就地逆置。

✧ 2.20 已知线性表中的元素以值递增有序排列，并以带头结点的单链表作存储结构。请写一个高效的算法，删除表中所有值相同的多余元素（使得操作后的线性表中的所有元素的值各不相同），同时释放被删除结点空间。

13

✧ 2.29 已知A，B和C为三个递增有序的线性表，先要求对A表作如下操作：删去那些既在B表中出现又在C表中出现的元素。试对顺序表编写实现上述操作的算法，并分析你的算法的时间复杂度(注意：题中没有特别指明同一表中的元素值各不相同)

14