

《数据结构与算法》第三次作业练习题

开始时间 09/22/2023 11:30:00 AM 结束时间 09/26/2023 11:59:00 PM 答题时长 6509分钟

答卷类型 标准答案 总分 40

判断题

得分：暂无 总分：10

1-1 线性表存储于A[1..50]中，删除第30个元素，需要移动20个元素。（ ）(2分)

☒ T ☐ F

1-2 顺序存储的空间必须是地址连续的一片空间，链式存储的空间地址可以连续也可以不连续。(2分)

☒ T ☐ F

1-3 在双向链表中，可以从当前结点出发访问到任何一个结点。(2分)

☒ T ☐ F

1-4 线性表L如果需要频繁地进行不同下标元素的插入、删除操作，此时选择顺序存储结构更好。(2分)

☐ T ☒ F

1-5 对单链表来说，只有从头结点开始才能访问到表中所有结点。(2分)

☒ T ☐ F

单选题

得分：暂无 总分：10

2-1 线性表有元素50个，存储在数组A【1..100】中，在第25个元素之后插入一个元素，需要移动元素（ ）个。(2分)

- ☐ A. 24
☒ B. 25
☐ C. 26
☐ D. 75

2-2 在N个元素的顺序表中，算法的时间复杂度为O(1)的是（ ）。(2分)

- ☐ A. 对N个元素排序
☐ B. 删除第1个元素
☒ C. 访问第K个元素 ($1 \leq K \leq N$)
☐ D. 在第1个元素之前插入1个元素

2-3 链表结点定义为(data,next),在P指向的结点之后插入结点S的代码是(2分)

- ☐ A. $P \rightarrow next = S; S \rightarrow next = P;$
☐ B. $P = S; S \rightarrow next = P \rightarrow next;$
☒ C. $S \rightarrow next = P \rightarrow next; P \rightarrow next = S;$
☐ D. $S = P; P \rightarrow next = S;$

2-4 使用双向链表存储数据，优点是可以(2分)

- ☐ A. 提高检索速度
☐ B. 节约存储空间
☒ C. 方便插入删除数据
☐ D. 能很快回收存储空间

2-5 单链表 - 时间复杂度(2分)

在包含 n 个数据元素的单链表中，_____的时间复杂度不为 $O(n)$ 。

- ☐ A. 访问第 $i (1 \leq i \leq n)$ 个数据元素
☐ B. 在位序 $i (1 \leq i \leq n + 1)$ 处插入一个新结点

☐ C. 删除位序 $i(1 \leq i \leq n)$ 处的结点☒ D. 将 n 个元素按升序排序

函数题

得分：暂无 总分：20

6-1 顺序表的删除 (10分)

给定一个顺序存储的线性表，请设计一个函数删除所有值小于min和值大于max的元素。删除后表中剩余元素保持顺序存储，并且相对位置不能改变。

函数接口定义：

```
List Delete( List &L, ElementType minD, ElementType maxD );
```

L是用户传入的一个线性表，其中ElementType元素可以通过>、==、<进行比较。函数Delete应将Data[]中所有值小于minD和值大于maxD的元素删除，同时保证表中剩余元素保持顺序存储，并且相对位置不变，最后返回删除后的表。

裁判测试程序样例：

```
#include <stdio.h>
#define MAXSIZE 20
typedef int ElementType;
typedef int Position;
typedef struct LNode *List;
struct LNode {
    ElementType Data[MAXSIZE];
    Position last; /* 线性表的长度*/
};

List Delete( List &L, ElementType minD, ElementType maxD );
List ReadInput();
{
    int num,i,temp;
    scanf("%d",&num);
    for(i=0;i<num;i++)
    {
        scanf("%d",&(L->Data[num]));
    }
    L->last = num-1;
}

void PrintList( List L );
{
    for(int i=0;i<=L->last;i++)
    {
        printf("%d ",L->Data[i]);
    }
}

int main()
{
    List L;
    ElementType minD, maxD;
    int i;
    L = ReadInput();
    scanf("%d %d", &minD, &maxD);
    L = Delete( L, minD, maxD );
    PrintList( L );
    return 0;
}
```

/* 你的代码将被嵌在这里 */

输入样例：

在这里给出一组输入。例如：

```
10
4 -8 2 12 1 5 9 3 3 10
0 4
```

输出样例：

在这里给出相应的输出。例如：

```
4 2 1 3 3
```

6-2 寻找链表元素的前驱结点 (10分)

本题要求实现一个函数，可查找单链表(无监督元)中某个元素的前驱结点。例如链表中的元素为1, 6, 3, 2, 4, 查找3的前驱为6。如果表中无此元素或没有前驱则返回空指针。

函数接口定义：

```
ptr pre (ptr h,int x);
```

其中 **h** 和 **x** 是用户传入的参数，**x** 为查找的元素。返回查找元素结点的地址。

pre结构定义：

```
typedef struct node//结构体定义
{
    int data;//存储数据
    struct node *next;//指向下一个结点的指针
}snode,*ptr;//定义别名
```

裁判测试程序样例：

```
#include <stdio.h>
typedef struct node//结构体定义
{
    int data;
    struct node *next;
}snode,*ptr;

ptr pre (ptr h,int x);

int main()
{
    ptr head,p;
    int x;
    head=creat();//构造链表，无需用户完成，细节不表
    scanf("%d",&x);
    p=pre(head,x);
    if(p==NULL)
        printf("None");
    else
        printf("%d",p->data);
    return 0;
}

/* 请在这里填写答案 */
```

输入样例：

5
1 6 3 2 4
3

输出样例：

6