

# 山西工程技术学院

## 实验报告

(2023 -2024 学年第 1 学期)

课程名称: 数据结构与算法实训

专业班级:

学 号:

学生姓名:

任课教师:

2023 年 11 月

实验名称	两个一元稀疏多项式相加			指导教师	
实验类型	验证型	实验学时	4	实验时间	11.7
<b>一、实验目的</b> 实验要求：多项式 a 和 b 相加, 建立单链表多项式 a+b, 和结果保存在 a, 销毁 b。 实验目的：掌握单链表的插入、删除、修改操作					
<b>二、实验环境</b> Dev C++;					
<b>三、实验内容</b> 1)输入并建立多项式 a 和 b 2)输出多项式,输出形式为整数序列:n,c1,e1,n2,e2,...cn,en,其中 n 是多项式的项数,ci 和 ei 是第 i 项的系数和指数,序列按指数升序排列 3)多项式 a 和 b 相加建立多项式 a+b  <pre> #include&lt;stdio.h&gt; #include&lt;stdlib.h&gt; #include&lt;string.h&gt; typedef struct polynomial {     int coef;//系数     int exp;//指数     struct polynomial *next; }*Link, Node; void inputPoly(Link head);//用于从控制台读入链表的函数 void print(Link head);//打印链表用的函数 bool insert(Link head, int coef, int exp);//向链表插入一个元素的函数 void combin2List(Link heada, Link headb, Link headab);//合并两个链表 int main() {     Link headA, headB;//两个多项式的头指针     Link headAB;//合并后的多项式的头指针      /*链表的初始化*/     headA=(Link)malloc(sizeof(Node));     headA-&gt;next=NULL;     headB=(Link)malloc(sizeof(Node));           </pre>					

```

headB->next=NULL;
headAB=(Link)malloc(sizeof(Node));
headAB->next=NULL;

printf("请输入第一个多项式的系数和指数, 以(0 0)结束: \n");
inputPoly(headA);
printf("第一个");
print(headA);
printf("请输入第二个多项式的系数和指数, 以(0 0)结束: \n");
inputPoly(headB);
printf("第二个");
print(headB);

combin2List(headA, headB, headAB);
printf("合并后");
print(headAB);
return 0;
}

void inputPoly(Link head)
{
    int coef, exp; //系数和指数
    printf("请输入系数和指数(如: \"2 3\"表示 2x^3): ");
    scanf("%d %d", &coef, &exp);
    while(coef!=0 || exp!=0) //连续输入多个系数和指数
    {
        insert(head, coef, exp); //调函数输入多项式
        printf("请输入系数和指数: ");
        scanf("%d %d", &coef, &exp);
    }
}

bool insert(Link head, int coef, int exp)
{
    Link node; //node 指针指向新创建的节点
    Link q, p;
    q = head;
    p = head->next;
    node = (Link)malloc(sizeof(Node)); //创建一个新结点
    if(head->next==NULL) //空表, 插第 1 个
    {
        node->coef = coef;
        node->exp = exp;
        head->next = node;
        node->next = NULL;
    }
}

```

```

void print(Link head)
{
    Link p; //指向链表要输出的结点
    printf("多项式如下: \n");
    p=head->next;

    if (p == NULL)
    {
        printf("多项式为\n");
        return;
    }
    // 不是空表
    char item[20]="";//要打印的当前多项式的一项
    char number[7]="";//暂时存放系数转换成的字符串

    bool isFirstItem=true;//标志是否为第一个节点的 flag
    //打印节点
    while(p != NULL)
    {
        if(isFirstItem) //如果是第一项，不要打+号
        {
            isFirstItem = false;
            if(p->coef > 0)
            {
                if(p->coef !=1) {
                    if(p->exp == 1) printf("%dx",p->coef);
                    printf("%dx^d",p->coef,p->exp);}
                else{
                    if(p->exp == 1) printf("x");
                    printf("x^d",p->exp);}//如果系数为1，不用打系数
            }
        }
        else if(p->coef < 0)//如果系数为负数，系数自身带有符号
        {
            if(p->coef != -1) {
                if(p->exp == 1) printf("%dx",p->coef);
                printf("%dx^d",p->coef,p->exp);}
            else{
                if(p->exp == 1) printf("-x");
                printf("-x^d",p->exp);}//系数为-1 打印负号
            }
        }
        else
            printf("");
    }
}

```

```

void combin2List(Link heada, Link headb, Link headab)
{
    Link pa, pb, pab; //指向 a, b 链表和 ab 的指针
    pa = heada->next;
    pb = headb->next;
    pab = headab->next;

    while (pa != NULL && pb != NULL) //a, b 链表都没有访问完毕
    {
        //如果指数 a > 指数 b, a 节点插入 ab 链表, a 指针后移
        if (pa->exp > pb->exp)
        {
            insert(headab, pa->coef, pa->exp);
            pa = pa->next;
        }
        //如果指数 a < 指数 b, b 节点插入 ab 链表, b 指针后移
        else if (pa->exp < pb->exp)
        {
            insert(headab, pb->coef, pb->exp);
            pb = pb->next;
        }
        //如果指数 a == 指数 b, a、b 系数相加, 插入 ab 链表, a、b 指针后移
        else if (pa->exp == pb->exp)
        {
            pa->coef += pb->coef;
            insert(headab, pa->coef, pa->exp);
            pa = pa->next;
            pb = pb->next;
        }
    }
    //如果 a、b 链表还有尾巴, 将它加到 ab 链表后面
    while (pa != NULL)

```

#### 四、实验小结

- (1) 学会了指针和数据结构的用法。
- (2) 使用了一些数学的知识, 需要有一定的逻辑思维和数学基础。

实验成绩		批阅日期		批阅人	
------	--	------	--	-----	--



实验名称	构造赫夫曼编码进行通信			指导教师	
实验类型	验证型	实验学时	4	实验时间	11.9

### 三、实验目的

给定字符和权值,设计每个字符的赫夫曼编码  
掌握如何构造一颗赫夫曼树的赫夫曼编码

### 四、实验环境

Dev C++;

### 三、实验内容

```
#include<iostream>
#include<string.h>
#include<malloc.h>
#define MAX 100
using namespace std;

typedef struct{
    int weighth;
    int parent;
    int rchild;
    int lchild;
}HTNode,*HuffmanTree;

static char N[MAX];

typedef char **HuffmanCode;

typedef struct{
    int s1;
    int s2;
}MinCode;

MinCode Select(HuffmanTree HT,int n);

HuffmanCode HuffmanCoding(HuffmanTree HT,HuffmanCode HC,int *w,int n){
    int i,s1=0,s2=0;
```

```

HuffmanTree p;

char *cd;
int f, c, start, m;

MinCode min;

m=2*n-1;

HT=(HuffmanTree)malloc((m+1)*sizeof(HTNode));

for(p=HT, i=0; i<=n; i++, p++, w++) {
    p->weight=*w;
    p->parent=0;
    p->rchild=0;
    p->lchild=0;
}

for(; i<=m; i++, p++) {
    p->weight=0;
    p->parent=0;
    p->lchild=0;
    p->rchild=0;
}

for(i=n+1; i<=m; i++) {

    min=Select(HT, i-1);
    s1=min.s1;
    s2=min.s2;

    HT[s1].parent=i;
    HT[s2].parent=i;
    HT[i].lchild=s1;
    HT[i].rchild=s2;
    HT[i].weight=HT[s1].weight+HT[s2].weight;//合并权值
}
cout<<"哈夫曼树列表:  "<<endl;
    cout<<"序号\t权值\t父节点\t左孩子\t右孩子\n";
    for(i=1; i<=m; i++)

printf("%d\t%d\t%d\t%d\t%d\n", i, HT[i].weight, HT[i].parent, HT[i].lchild, HT[i].rchild)
;

```



```

cd=(char *)malloc(n*sizeof(char *));
cd[n-1]='\0';
    for(i=1;i<=n;i++) {
        start=n-1;

        for(c=i, f=HT[i].parent; f!=0; c=f, f=HT[f].parent) {

            if(HT[f].lchild==c)
                cd[--start]='0';
            else
                cd[--start]='1';
        }

        HC[i]=(char *)malloc((n-start)*sizeof(char *));

        strcpy(HC[i], &cd[start]);
    }
    free(cd);
    return HC;
}

```

//选出最小与次小的函数

```

MinCode Select(HuffmanTree HT, int n) {
    int min, secmin;
    int i, s1, s2;
    MinCode code;
    s1=1;
    s2=1;
    min=0x3f3f3f3f;
    for(i=1; i<=n; i++) {
        if(HT[i].weight<min && HT[i].parent==0) {
            min=HT[i].weight;
            s1=i;
        }
    }
    secmin=0x3f3f3f3f;
    for(i=1; i<=n; i++) {
        if((HT[i].weight<secmin) && (i!=s1) && HT[i].parent==0) {
            secmin=HT[i].weight;
            s2=i;
        }
    }
    code.s1=s1;
    code.s2=s2;
    return code;
}

```

```

int main() {
    HuffmanTree HT=NULL;
    HuffmanCode HC=NULL;
    int *w=NULL;
    int i,n;
    cout<<"请输入要编码的字符串：";
    gets(N);
    n=strlen(N);
    w=(int *)malloc((n+1)*sizeof(int *));
    w[0]=0;
    cout<<"依次输入权值："<<endl;
    for(i=1;i<=n;i++) {
        printf("w[%d]=",i);
        scanf("%d",&w[i]);
    }
    HC=HuffmanCoding(HT,HC,w,n);
    cout<<"哈夫曼编码："<<endl;
    cout<<"字符 权值 编码";
    for(i=1;i<=n;i++)
        cout<<N[i-1]<<w[i]<<HC[i];
    cout<<"哈夫曼编码："<<endl;
    for(i=1;i<=n;i++)
        printf("%s\t",HC[i]);
    cout<<endl;
    return 0;
}

```

#### 四、实验小结

学会使用 C++ 编写哈夫曼树  
理解和使用哈夫曼编码和哈夫曼树

实验成绩		批阅日期		批阅人	
------	--	------	--	-----	--



实验名称	设计一个校园导游咨询			指导教师	
实验类型	验证型	实验学时	4	实验时间	7.14
<b>一、实验目的</b> 给定一个图有 10 个顶点，代表校园的 10 个景点，找一颗最小生成树可以把这 10 个景点都游览，但路径最短。 掌握无向连通图中最小生成树的生成					
<b>二、实验环境</b> Dev C++;					
<b>三、实验内容</b> <pre> #include &lt;iostream&gt;  #include &lt;stdlib.h&gt; #include &lt;windows.h&gt; using namespace std; #define MaxVertexNum 100 #define MaxCost 9999 typedef char VertexType; typedef int EdgeType; typedef struct { VertexType vexs[MaxVertexNum];   /* 存放顶点信息 */   EdgeType edges[MaxVertexNum][MaxVertexNum];   /* 存放邻接关系 */   int n,e; /*顶点数和边数*/ }Mgraph; void mune(); //主菜单界面 void floyd(Mgraph G,int D[MaxVertexNum][MaxVertexNum],int P[MaxVertexNum][MaxVertexNum]); //指定距离的弗洛伊德 void opfloy(int n,int D[MaxVertexNum][MaxVertexNum],int P[MaxVertexNum][MaxVertexNum]); //指定距离的输出 void hamanzi(Mgraph *G); //从起点开始遍历的函数。 void CreateMGraph(Mgraph *G); //建立图的函数 void printph(); void look(); int main() { </pre>					

```

Mgraph G;
    int n,D[MaxVertexNum][MaxVertexNum],P[MaxVertexNum][MaxVertexNum];
    int a;
    char b[100];
    char user[20],mana[20]="password",*q,*p,m=0;//依次是登录者，管理者，指针来判断
    登录账号是否是管理者。
    system("color f0");
L: printf("\t      *****\n");
    printf("\t 请登录（输入你的用户名）:");
    scanf("%s",user);
    p=user;
    q=mana;
    while(*p==*q)//用来判断登陆者是否为管理者的部分
    {
        m=1;
        p++;
        q++;
        if(*q=='\0')
            break;
    }
    if(*p!=*q)
        m=0;
    /*****/
    if(m==1)
    {
        printf("");

        CreateMGraph(&G);
        system("cls");
        L1: mune();
        printf("请输入您的选项\n");
        scanf("%d",&a);
        switch(a)
        {
            case 1:
                floyd(G,D,P);
                opfloy(G.n,D,P);
                system("pause");
                system("cls");
                a=35421568;
                goto L1;
            case 2:
                hamanzi(&G);
                system("pause");

```

```

a=35421568;
    goto L1;
case 3:
    printph();
    system("pause");
    system("cls");
    a=35421568;
    goto L1;
case 4:
    look();
    system("cls");
    a=35421568;
    goto L1;
case 0:
    break;

default:
    printf("您输入有误, 请重新输入!!!\n");
    system("pause");
    gets(b);
    a=35421568;
    system("cls");
    goto L1;

    }
}
else
{
//CreateMGraph(&G);
    system("cls");
    L3:
    printf("\t\t\t 您不是管理者无法对景点信息进行修改.");

printf("*****\n");
    printf("\t\t\t-----校园导游系统-----\n");
    printf("\t\t\t 1: 去指定的景点\n");
    printf("\t\t\t 2: 遍历全部景点\n");
    printf("\t\t\t 3: 查看校园的全图\n");
    printf("\t\t\t 4: 查看景点介绍\n");
    printf("\t\t\t 5: 返回登陆界面\n");
    printf("\t\t\t 0: 退出\n");

```

```

printf("*****\n");
printf("请输入您的选项\n");
scanf("%d",&a);
switch(a)
{
    case 1:
        floyd(G,D,P);
        opfloy(G.n,D,P);
        system("pause");
        system("cls");
        a=35421568;
        goto L3;
    case 2:
        hamanzi(&G);
        system("pause");
        system("cls");
        a=35421568;
        goto L3;
    case 3:
        printph();
        system("pause");
        system("cls");
        a=35421568;
        goto L3;
    case 4:
        look();
        system("cls");
        a=35421568;
        goto L3;
    case 5:
        a=35421568;
        system("cls");
        goto L;
    case 0:
        break;

    default: printf("您输入有误, 请重新输入!!!\n");
        system("pause");
        gets(b);
        a=35421568;
        system("cls");
        goto L3;
}

```

```

    }
}

return 0;
}

void mune()
{
printf("*****\n");
printf("\t\t\t-----校园导游系统-----\n");
printf("\t\t\t 1:去指定的景点\n");
printf("\t\t\t 2:遍历全部景点\n");
printf("\t\t\t 3:查看校园的全图\n");
printf("\t\t\t 4:查看景点介绍\n");
printf("\t\t\t 0:退出\n");

printf("*****\n");
}

void floyd(Mgraph G,int D[MaxVertexNum][MaxVertexNum],int
P[MaxVertexNum][MaxVertexNum])
{ int i,j,k;
for(i=0;i<G.n;i++)
for(j=0;j<G.n;j++)
{D[i][j]=G.edges[i][j];
if ((D[i][j]<MaxCost)&&(D[i][j]!=0))
P[i][j]=j;
else
P[i][j]=-1;
}
for(i=0;i<G.n;i++)
D[i][i]=0;
for(k=0;k<G.n;k++)
for(i=0;i<G.n;i++)
for(j=0;j<G.n;j++)
if (D[i][j]>D[i][k]+D[k][j])
{D[i][j]=D[i][k]+D[k][j];
P[i][j]=P[i][k];
}
}

void opfloy(int n,int D[MaxVertexNum][MaxVertexNum],int
P[MaxVertexNum][MaxVertexNum])
{int i,j,k,c,m,p,min,next;
printf("请输入你现在的位置和将去的位置加(m,n)\n");

```



```

scanf("%d,%d",&m,&p);
    i=m;
    j=p;
    // for (i=0;i<n;i++)
    // printf("\n\n 源点为 v%d:", i);
    // for (j=0;j<n;j++)
    if (j!=i)
    {printf("\n%d 到%d 应按如下行走 ,   最近距离为:\n", i, j);
        printf("%d", i);
        next=P[i][j];
        while (next!=-1)
        {printf("->%d", next);
            next=P[next][j]; }
        if (D[i][j]==MaxCost)
            printf("->%d", j);
        printf("\t%d", D[i][j]); }
    }
void hamanzi(Mgraph *G)
{
    int i, j, m, wei[5], sum=0, min=100, end[5], zui[6];
    int q, w, t, r;

    for(i=1; i<5; i++)
    {
        end[i]=0;
    }
    printf("请输入你的起点。 \n");
    scanf("%d",&m);
    for(q=0; q<5; q++)
    {
        end[m]=1;
        end[q]=1;
        for(w=0; w<5; w++)
        { end[w]=1;
            for(t=0; t<5; t++)
            {
                end[t]=1;
                for(r=0; r<5; r++)
                {

```

```

end[r]=1;

sum=G->edges[m][q]+G->edges[q][w]+G->edges[w][t]+G->edges[t][r]+G->edges[r][m];
    end[m]=1;
    end[q]=1;
    end[w]=1;
    end[t]=1;
    end[r]=1;

    if(min>sum&&end[0]==1)
    {
        if(end[1]==1&&end[2]==1)
        {
            if(end[3]==1&&end[4]==1)
            {
                min=sum;
                zui[1]=q;
                zui[2]=w;
                zui[3]=t;
                zui[4]=r;
                zui[5]=m;
            }
        }
    }
    sum=200;
    end[r]=0;

    end[t]=0;
    }
    end[w]=0;
    }
    end[q]=0;
}
printf("最小为: %d\n",min);
printf("应按照如下浏览: \n");
printf("%d",m);
for(i=1;i<6;i++)
{
    printf("->%d",zui[i]);
}
}

```

```

void CreateMGraph(Mgraph *G)
{
    int i, j, k, w;
    printf("请输入顶点数和边数(输入格式为:顶点数, 边数):\n") ;
    scanf("%d,%d",&(G->n), &(G->e));
    printf("请输入顶点信息:\n");
    for(i=0;i<G->n;i++)
        scanf("\n%c",&(G->vexs[i]));
    for(i=0;i<G->n;i++)
        for(j=0;j<G->n;j++)
            G->edges[i][j]=MaxCost;
    printf("请输入每条边对应的两个顶点的序号(输入格式为:i, j, w):\n");
    for(k=0;k<G->e;k++)
    {
        scanf("%d,%d,%d",&i,&j,&w);
        G->edges[i][j]=w;
    }
}

void printph()
{
    printf("公园地图:\n");
    printf("\t\t\t\t0* * * * * 1\n");
    printf("\t\t\t\t**          **\n");
    printf("\t\t\t\t* *          * *\n");
    printf("\t\t\t\t* *          * *\n");
    printf("\t\t\t\t* *          * *\n");
    printf("\t\t\t\t* *          * *\n");
    printf("\t\t\t\t* 2          *\n");
    printf("\t\t\t\t*          *\n");
    printf("\t\t\t\t*          *\n");
    printf("\t\t\t\t*          *\n");
    printf("\t\t\t\t*          *\n");
    printf("\t\t\t\t* *          *\n");
    printf("\t\t\t\t4* * * * * 3\n");
}

void look()
{
    int a;
    char b[100];
    system("cls");
    L2:
    printf("\t\t\t\t1:能源之火\n");
    printf("\t\t\t\t2:图书馆\n");
    printf("\t\t\t\t3:勤学楼\n");
    printf("\t\t\t\t4:大数据中心\n");
    printf("\t\t\t\t5:西操场\n");
}

```

```

printf("\t\t\t\t 6:思齐楼\n");
printf("\t\t\t\t 7:学致苑\n");
printf("\t\t\t\t 8:羽乒馆\n");
printf("\t\t\t\t 9:双创中心\n");
printf("\t\t\t\t 10:宿舍\n");
printf("\t\t\t\t 0:退回主界面\n");
printf("请输入您的选择.\n");
scanf("%d",&a);
switch(a)
{
    case 1:
        printf("能源之火, 象征着山西工程技术学院有关煤的历史");
        system("pause");
        system("cls");
        a=65944899;
        goto L2;
    case 2:
        cout<<"图书馆是新建立的, 有这非常多的藏书, 覆盖所有学科的专业课程的书目";
        system("cls");
        a=65944899;
        goto L2;
    case 3:
        printf("勤学楼是学生们上专业课的地方, 全部配备多媒体, 用来方便同学们的学习");
        system("pause");
        system("cls");
        a=65944899;
        goto L2;
    case 4:
        printf("大数据中心有配备有 10 台服务器, 用来支持校园网和一些学校相关的信息资源");
        system("pause");
        system("cls");
        a=65944899;
        goto L2;
    case 5:
        printf("西操场是一个标准的 400 米操场给同学们提供了最好的运动和锻炼空间" 。 \n");
        system("pause");
        system("cls");
        a=65944899;
        goto L2;
}

```

```

case 6:
    printf("思齐楼，配备完善的大学物理实验室" 。\n");
    system("pause");
    system("cls");
    a=65944899;
    goto L2;
case 7:
    printf("学知园餐厅，提供一流的餐饮服务" 。\n");
    system("pause");
    system("cls");
    a=65944899;
    goto L2;
case 8:
    printf("羽乒馆，为同学们提供休闲娱乐场所" 。\n");
    system("pause");
    system("cls");
    a=65944899;
    goto L2;
case 9:
    printf("双创中心，配备省内一流的教学资源" 。\n");
    system("pause");
    system("cls");
    a=65944899;
    goto L2;
case 10:
    printf("学生宿舍，提供优质住宿环境" 。\n");
    system("pause");
    system("cls");
    a=65944899;

```

## 五、实验小结

学会熟练使用图和最小生成树

实验成绩		批阅日期		批阅人	
------	--	------	--	-----	--

