南京信息工程大学 数据结构 实验(实习)报告

实验(实习)名称  **线性表** 实验(实习)日期 2021.10.21 得分 指导老师 马瑞

系**计算机与软件学院**专业 **计算机科学与技术**班级 20(1) 姓名 鲁哲豪 学号 202083290400

一、实验目的



二、 实验内容与步骤







1.

先malloc一个新的顺序表，空间是原来的两个顺序表相加。使用rand函数随机生成10个数作为测试数据。使用qsort函数进行排序，以满足题目的初始条件。

在升序的情况下，依次访问两个顺序表第1个元素，将较小的那个加入新的顺序表。

重复上一步骤，直到两个顺序表皆为空。注意当一个顺序表为空的时候，直接将另一顺序表示剩下所有元素，依次加入新顺序表即可。

    for(i=0,j=0;i<a.len||j<b.len;r++){

        if(i==a.len){

            \*r=\*(q++);

            j++;

        }else if(j==b.len){

            \*r=\*(p++);

            i++;

        }else if(\*p<\*q){

            \*r=\*(p++);

            i++;

        }

        else{

            \*r=\*(q++);

            j++;

        }

    }

    return c;

}

#include<stdio.h>

#include<stdlib.h>

typedef struct{

    int \*base,len,size;

} List;

int cmp(const void\*a,const void\*b){

    return \*(int\*)a-\*(int\*)b;

}

void showList(List a){

    int i,\*p=a.base;

    for(i=0;i<a.len;i++)

        printf("%d ",\*(p++));

    printf("\n");

}

List mergeList(List a,List b){

    List c;

    c.len=a.len+b.len;

    c.size=a.size+b.size;

    c.base=malloc(c.size);

    int i,j,\*p=a.base,\*q=b.base,\*r=c.base;

    b.len=10;

    b.size=10\*sizeof(b.len);

    b.base=malloc(b.size);

    for(i=0;i<10;i++)

        (b.base)[i]=rand();

    qsort(b.base,b.len,sizeof(int),cmp);

    showList(b);

    List c=mergeList(a,b);

    showList(c);

    return 0;

}

int main(){

    srand(1);

    List a,b;

    a.len=10;

    a.size=10\*sizeof(a.len);

    a.base=malloc(a.size);

    int i;

    for(i=0;i<10;i++)

        (a.base)[i]=rand();

    qsort(a.base,a.len,sizeof(int),cmp);

    showList(a);

随机数种为1时生成的两个顺序表,以及合并的结果。

a: 41 6334 11478 15724 18467 19169 24464 26500 26962 29358

b: 491 2995 4827 5436 5705 9961 11942 16827 23281 28145

c: 41 491 2995 4827 5436 5705 6334 9961 11478 11942 15724 16827 18467 19169 23281 24464 26500 26962 28145 29358

2.

使用rand函数随机生成10个递增的整数作为测试数据，构建两个单链表。以满足题目的初始条件。

在升序的情况下，依次访问两个表头，将较小的那个加入新的表，并将头指针指向下一位。

重复上一步骤，直到一个单链表为空的时候，直接将另一单链的表头接到新单链表尾即可。

代码：

int main(){

    srand(1);

    Node \*a,\*b,\*p;

    int i,n=rand()/1000;

    a=malloc(sizeof(Node));

    a->data=n;

    a->next=NULL;

    p=a;

    for(i=0;i<9;i++){

        p=p->next=malloc(sizeof(Node));

        p->data=n+=rand()/1000;

        p->next=NULL;

    }

    show(a);

    n=rand()/1000;

    b=malloc(sizeof(Node));

    b->data=n;

    b->next=NULL;

    p=b;

    for(i=0;i<9;i++){

        p=p->next=malloc(sizeof(Node));

        p->data=n+=rand()/1000;

        p->next=NULL;

    }

    show(b);

    Node\* c=merge(a,b);

    show(c);

    return 0;

}

#include<stdio.h>

#include<stdlib.h>

typedef struct \_Node{

    int data;

    struct \_Node\* next;

}Node;

void show(Node\* p){

    do{

        printf("%d ",p->data);

    }while(p=p->next);

    printf("\n");

}

Node\* merge(Node\*a,Node\*b){

    Node \*c=malloc(sizeof(Node)),

        \*head=c;

    while(a && b){

        if(a->data < b->data){

            c=c->next=a;

            a=a->next;

        }else{

            c=c->next=b;

            b=b->next;

        }

        c->next=NULL;

    }

    if(a) c->next=a;

    else c->next=b;

    return head->next;

}

随机数种为1时生成的两个表,以及合并的结果：

