2018302141224-吴健豪

//老师好，您可以搜索“question”+题号进行定位

//question1

//Btree.h

#pragma once

#include <iostream>

#include<vector>

using namespace std;

struct BTNode

{

int data;

struct BTNode \*lchild, \*rchild;

};

BTNode \*CreateBTree(int a[], int b[], int n) //使用先序序列和中序序列

{

int k;

if (n <= 0)

return NULL;

BTNode \*bt = (BTNode \*)malloc(sizeof(BTNode));

bt->data = a[0];

for (k = 0; k < n; k++)

if (a[0] == b[k])

break;

//递归创建左子树

bt->lchild = CreateBTree(a + 1, b, k);

//递归创建右子树

bt->rchild = CreateBTree(a + k + 1, b + k + 1, n - 1 - k);

return bt;

}

void DispBTree(BTNode \*bt)

{

if (bt != NULL)

{

cout << bt->data << " ";

if (bt->lchild!=NULL||bt->rchild!=NULL)

{

cout << "(";

DispBTree(bt->lchild);

if (bt->rchild!=NULL)

{

cout << ",";

}

DispBTree(bt->rchild);

cout << ")";

}

}

}

void DestroyBTree(BTNode \*bt)

{

if (bt == NULL) {

return;

}

else {

DestroyBTree(bt->lchild);

DestroyBTree(bt->rchild);

free(bt);

}

}

//question1.cpp

#include "Btree.h"

int sum(BTNode \* bt)

{

if (bt == NULL)

{

return 0;

}

if (bt->lchild == NULL && bt->rchild == NULL)

{

return bt->data;

}

return (sum(bt->lchild) + sum(bt->rchild));

}

int main()

{

cout << "please enter the number of vector" << endl;

int n;

cin >> n;

vector<int> a(n), b(n);

cout << "please enter vector a" << endl;

for (int i = 0; i < n; i++)

{

cin >> a[i];

}

cout << "please enter vector b" << endl;

for (int i = 0; i < n; i++)

{

cin >> b[i];

}

BTNode \* bt;

bt = CreateBTree(&a[0], &b[0], n);

DispBTree(bt);

cout << sum(bt);

system("pause");

}

//question2

//question2.cpp

#include "Btree.h"//https://blog.csdn.net/ljp1919/article/details/47804695 利用此文的讲解进行了对同一个解决方案中的另一个项目中的头文件Btree.h的引用

int countnum(BTNode \* bt, int k)

{

if (bt==NULL)

{

return 0;

}

else

{

int lnum = countnum(bt->lchild, k);

int rnum = countnum(bt->rchild, k);

if (bt->data>=k)

{

return lnum + rnum + 1;

}

else

{

return lnum + rnum;

}

}

}

int main()

{

cout << "please enter the number of vector" << endl;

int n;

cin >> n;

vector<int> a(n), b(n);

cout << "please enter vector a" << endl;

for (int i = 0; i < n; i++)

{

cin >> a[i];

}

cout << "please enter vector b" << endl;

for (int i = 0; i < n; i++)

{

cin >> b[i];

}

BTNode \* bt;

bt = CreateBTree(&a[0], &b[0], n);

DispBTree(bt);

cout << endl;

cout << "please enter the k value" << endl;

int k;

cin >> k;

cout << countnum(bt,k);

system("pause");

}

//question3

//question3.cpp

#include "Btree.h"//https://blog.csdn.net/ljp1919/article/details/47804695 利用此文的讲解进行了对同一个解决方案中的另一个项目中的头文件Btree.h的引用

int findnode(BTNode \* bt, int x,int level)//level represents the present level

{

level++;

if (bt==NULL)

{

return 0;

}

else

{

if (bt->data==x)

{

return level;

}

else

{

int lnum = findnode(bt->lchild, x, level);

int rnum = findnode(bt->rchild, x, level);

return lnum >= rnum ? lnum : rnum;

}

}

}

int main()

{

cout << "please enter the number of vector" << endl;

int n;

cin >> n;

vector<int> a(n), b(n);

cout << "please enter vector a" << endl;

for (int i = 0; i < n; i++)

{

cin >> a[i];

}

cout << "please enter vector b" << endl;

for (int i = 0; i < n; i++)

{

cin >> b[i];

}

BTNode \* bt;

bt = CreateBTree(&a[0], &b[0], n);

DispBTree(bt);

cout << endl;

cout << "please enter the x value" << endl;

int x;

cin >> x;

cout << findnode(bt, x,0);

system("pause");

}

//question4

//question4.cpp

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int find\_i(int a[],int head, int tail)

{

if (a[head]==head)

{

return head;

}

if (a[tail]==tail)

{

return tail;

}

if (head==tail)

{

cout << "the vector does not have a element whose value equals to its label number" << endl;

return NULL;

}

int mid = 0.5\*(head + tail);

if (mid==head)

{

cout << "the vector does not have a element whose value equals to its label number" << endl;

return NULL;

}

if (a[mid]==mid)

{

return mid;

}

else

{

if (a[mid]>mid)

{

return find\_i(a, head, mid);

}

else

{

return find\_i(a, mid, tail);

}

}

}

int main()

{

cout << "please enter the number of the array" << endl;

int n;

cin >> n;

vector<int> a(n);

cout << "please enter the n different integers" << endl;

for (int i = 0; i < n; i++)

{

int value;

cin >> value;

a[i] = value;

}

sort(&a[0],&a[n-1]+1);//I found that the second parameter of STL's sort needs to be the last address plus 1(and &a[n] is false while &a[n-1]+1 is right)

cout << "the vector has been sorted already:" << endl;

for (int i = 0; i < n; i++)

{

cout << a[i]<<" ";

}

cout << endl << "the label is:" << endl<<find\_i(&a[0],0,n-1);

system("pause");

}

//question5

//question5.cpp

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int threefold(int a[],int head, int tail, int s)//for the second ask in this question, I would like to answer"log3(n)" as the time complexity for this algorithm.

{

if (tail-head<=2)

{

for (int i = head; i <= tail; i++)

{

if (a[i]==s)

{

return i;

}

}

return NULL;

}

int OneThird = head + (tail - head) / 3;

int TwoThirds = head + 2 \* (tail - head) / 3;

if (a[OneThird]==s)

{

return OneThird;

}

if (a[TwoThirds]==s)

{

return TwoThirds;

}

if (a[OneThird]>s)

{

return threefold(a, head, OneThird, s);

}

if (a[TwoThirds]>s)

{

return threefold(a, OneThird, TwoThirds, s);

}

return threefold(a, TwoThirds, tail, s);

}

int main()

{

cout << "please enter the number of the array" << endl;

int n;

cin >> n;

vector<int> a(n);

cout << "please enter the n different integers" << endl;

for (int i = 0; i < n; i++)

{

int value;

cin >> value;

a[i] = value;

}

sort(&a[0], &a[n - 1] + 1);//I found that the second parameter of STL's sort needs to be the last address plus 1(and &a[n] is false while &a[n-1]+1 is right)

cout << "the vector has been sorted already:" << endl;

for (int i = 0; i < n; i++)

{

cout << a[i] << " ";

}

cout << "please enter the value you are searching for:" << endl;

int searchnum;

cin >> searchnum;

cout << "the label number of the searchnum is:" << endl<<threefold(&a[0],0,n-1,searchnum);

system("pause");

}

//question6

//FindDecompositionFormula.cpp

#include <iostream>

using namespace std;

int solve(int n)

{

if (n==1)

{

return 1;

}

else

{

int sum = 0;

for (int i = 2; i <= n; i++)

{

if (n%i==0)

{

sum += solve(n / i);

}

}

return sum;

}

}

int main()

{

cout << "please enter the number" << endl;

int num;

cin >> num;

cout << "the result is: " << solve(num) << endl;

system("pause");

}