TRƯỜNG ĐẠI HỌC GIAO THÔNG VÂN TẢI KHOA CÔNG NGHỆ THÔNG TIN



---------------o0o---------------

# Bài tập lớn môn học

**CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

*Giảng viên hướng dẫn:* **TS. Hoàng Văn Thông** *Nhóm thực hiện:* 12 - Lớp CNTT 2 - K59

Danh sách sinh viên tham gia xếp theo thứ tự đóng góp vào báo cáo:

|  |  |  |
| --- | --- | --- |
| **STT** | **Mã SV** | **Họ tên** |
| 1 | 181200551 | Nguyễn Đình Bình |
| 2 | 181202600 | Đỗ Văn Hải |
| 3 | 181202287 | Lê Minh Hiếu |

## Hà Nội tháng 11 năm 2019

# I.Đề bài

Viết chương trình thực hiện các công việc sau trên một danh sách các số (***sử dụng cấu trúc dữ liệu danh sách liên kết đơn để lưu trữ).***

1. Tạo danh sách số, quá trình nhập sẽ dừng lại khi nhập dấu “#”.
2. Thêm một phần tử vào danh sách, vị trí thêm vào do người dùng chọn.
3. Nhập vào một số k (k0), đếm xem trong dãy có bao nhiêu số có giá trị = k?
4. Kiểm tra xem trong danh sách có 3 số chẵn dương đứng cạnh nhau hay không? Nếu có hãy in ra vị trí của các số này.
5. Sắp xếp danh sách theo thứ tự tăng dần.
6. Xóa tất cả các số nguyên tố trong danh sách.
7. Xóa tất cả các phần tử có giá trị trùng nhau trong danh sách, chỉ giữ lại 1 số.

II.Phân tích bài toán

1.Yêu cầu

-Sử dụng cấu trúc dữ liệu danh sách liên kết đơn để lưu trữ

-Viết chương trình thực hiện giải quyết bài toán đề bài trên một danh sách các số

2.Phân tích từng lớp

a.các lớp, thuộc tính, mô tả chức năng từng lớp từng phương thức

Các lớp

-Node

// Node là mắt xíc của SL , gồm giá trị và địa chỉ của node tiếp theo

-SingleList(SL)

//SL là danh sách liên kết đơn lưu trữ mảng, ds các số

-SinglistITR

//SL\_ITR bộ lặp duyệt SL

-NumberList

//App chức năng sử dụng để giải quyết bài toán

-Main(hàm main để chạy)

=============================

Node

Các thuộc tính

T Elem;

Node \*next;

Các phương thức của lớp

Node();

Void setNext(Node \*p);

Void setElem(T e);

T getElem();

Node \*getNext();

==============================

SingleList

Các thuộc tính:

Node \* header;

Node \* trailer;

Các phương thức của lớp

Singlelist();

Long size();

Int isEmpty();

Node \*first();

Node \*last();

void replace(Node<T> \*p, T e);

void remove(Node<T> \*p);

Node<T> \*insertAfter(Node<T>\*p, T e);

Node<T> \*insertFirst(T e);

Node<T> \*insertLast(T e);

Node<T> \*getNode(int i);

void sort();

void delList();

============================

Numberlist

Các thuộc tính

singleList<ll> v;

Các phương thức

int menu();

//menu hiển thị giao diện thao tác nhập,xuất, giải quyết với các chức năng

void run();

//hàm chạy chương trình

void InputArray();

//chức năng 1 tạo ds số, dừng nhập khi gặp #

void Insert();

//thêm 1 pt vào ds , vị trí thêm vào do người dùng chọn

void CountValue();

//Nhập vào một số k đếm xem trong dãy có bao nhiêu phần tử

void ThreeEvenNumber();

//kiểm tra trong ds có 3 số chẵn đứng cạnh nhau

void SortInc();

//Sắp xếp ds theo thứ tự tăng

void DelPrime();

//Xóa tất cả các số nguyên tố trong ds

void DelDuplicate();

//xóa tất cả các phần tử có giá trị trùng giữ lại 1 số

void OutPut();

//in ra ds liên kết

friend int PrimeNumber(ll);

//hàm kiểm tra nguyên tố đúng trả về 1, sai trả về 0

friend int even(ll);

//kiểm tra số chẵn dương đúng 1 sai trả về 0

===========

SLItr

Các thuộc tính

singleList<T> \*slist;

Node<T> \*curNode;

Các phương thức

singleListItr(singleList<T> \*list)

void resest()

int hasnext()

T next()

b. độ phức tạp(phân tích chi tiết ở dưới-cài đặt các lớp và hàm main)

|  |  |  |
| --- | --- | --- |
| Phương thức | Tính được | Độ phức tạp |
| int menu(); | 2 | O(1) |
| void run(); | =dpt từng hàm con | ? |
| void InputArray(); | 22 | O(1) |
| void Insert(); | 5n+k | O(n) |
| void CountValue(); |  | O(n) |
| void ThreeEvenNumber(); | 2 vong for | O(n^2) |
| void SortInc(); |  | O(n) |
| void DelPrime(); |  | O(n^2) |
| void DelDuplicate(); | 2 vong for | O(n^2) |
| void OutPut(); | 2 vong for | O(n^2) |
| friend int PrimeNumber(ll); | 2 vong for | O(n) |
| friend int even(ll); | 3 | O(1) |

III Cài đặt các lớp và hàm main bằng C++

#include"node.cpp"

#include"singlelist.cpp"

#include"singlelistitr.cpp"

using namespace std;

typedef long long ll;

class NumberList

{

private:

singleList<ll> v;

public:

int menu();

void run();

void InputArray();

void Insert();

void CountValue();

void ThreeEvenNumber();

void SortInc();

void DelPrime();

void DelDuplicate();

void OutPut();

friend int PrimeNumber(ll);

friend int even(ll);

};

int NumberList::menu()

{

cout<<"1. Nhap day (nhap dau # de dung).";

cout<<"\n2. Chen phan tu.";

cout<<"\n3. Dem gia tri k.";

cout<<"\n4. Kiem tra 3 so chan dung canh nhau (neu co se in vi tri).";

cout<<"\n5. Sap xep danh sach tang dan.";

cout<<"\n6. Xoa tat ca so nguyen to.";

cout<<"\n7. Xoa nhung phan tu trung nhau va giu lai 1 phan tu.";

cout<<"\n8. In danh sach.";

cout<<"\n9. Thoat";

cout<<"\nNhap thao tac: ";

int select;//->dpt:1

cin >> select;

return select; //->dpt:1

}

=>dpt O(1)

void NumberList::run()

{

int sel; //->dpt:1

do

{

system("cls");//dpt 1

sel = menu();//dpt 3+1

system("cls");//dpt 1

switch(sel)//

{

case 1:

InputArray();

break;

case 2:

Insert();

break;

case 3:

CountValue();

break;

case 4:

ThreeEvenNumber();

break;

case 5:

SortInc();

break;

case 6:

DelPrime();

break;

case 7:

DelDuplicate();

break;

case 8:

OutPut();

break;

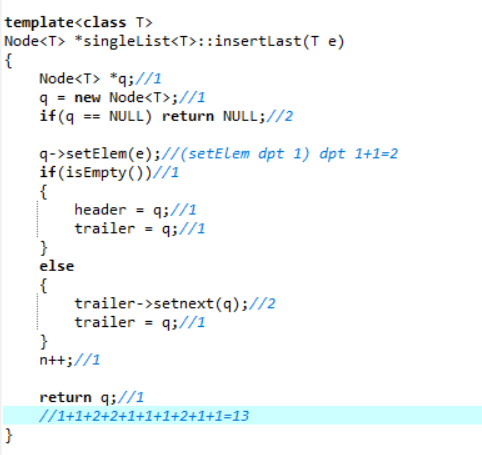
}

}

while(sel != 9);

}

Có đpt insertLast là O(1)



void NumberList::InputArray()

{

ll e; //dpt 1

char c[100]; //dpt 1

Node<ll> \*p; //dpt 1

while(1)

{

cout<<"\nNhap gia tri: ";

cin >> c; //dpt 1

if(c[0] == '#') //dpt 2

{

cout << "\nKet thuc qua trinh nhap!\n";

break;

}

sscanf(c, "%lld", &e); //dpt 1

p = v.insertLast(e);//dpt 13

if (p) cout << "\nChen thanh cong!";//1

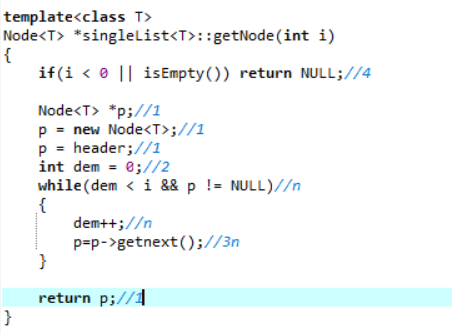
else cout<<"\nKhong the chen them!";

}

system("pause");//1

}

Ta có đpt của getNode là 5n+10 O(n)



Và đpt insert after : 16

void NumberList::Insert()

{

int i;//1

ll e;//1

Node<ll> \*p;//1

cout << "\nNhap gia tri phan tu: ";

cin >> e;//1

cout << "\nNhap vi tri muon chen: ";

cin >> i;//1

if(i == 0)//1

{

if(v.insertFirst(e)) cout << "\nChen thanh cong!";//13

else cout << "\nLoi chen!\n";

}

else

{

p = v.getNode(i - 1);//3+5n +10

if (v.insertAfter(p,e)) cout << "\nChen thanh cong!";//16

else cout << "\nLoi chen!\n";

}

system("pause");//1

}

void NumberList::CountValue()

{

if(!v.isEmpty())

{

ll k, dem = 0;

do

{

cout << "\nNhap gia tri khac 0: ";

cin >> k;

if(!k) cout << "\nLoi. Nhap lai!";

}

while(!k);

Node<ll> \*p;

p = v.first();

while(p != NULL)

{

if(p->getElem() == k) dem++;

p = p->getnext();

}

cout << "\nGia tri " << k << " co " << dem << " phan tu.";

}

else cout << "\nDanh sach trong!";

system("pause");

}

int even(ll a)

{

return (!(a % 2 ) && a > 0);

}

void NumberList::ThreeEvenNumber()

{

if(!v.isEmpty())

{

int count = 0, index = 0, check = 0;

for(Node<ll>\*p = v.first(); p != NULL; p = p->getnext())

{

if(even(p->getElem())) count++;

else count = 0;

if(count >= 3)

{

check = 1;

cout << "\nVi tri 3 so chan dung canh nhau: ";

cout << index - 2 << " " << index - 1 << " " << index << endl;

}

index++;

}

if(!check) cout << "\nKhong co 3 phan tu chan duong nao dung canh nhau!";

}

else cout << "\nDanh sach trong!";

system("pause");

}

int PrimeNumber(ll a)

{

if (a < 2) return 0;//2

if (a == 2) return 1;//2

if (!(a % 2)) return 0;//1+1+1=3

ll root;//1

root = sqrt((double)a);//1+1+1

for(int i = 3; i <= root; i += 2 )//1+1

{

if (!(a % i)) return 0;

}

return 1;

}

void NumberList::DelPrime()

{

singleList<ll> New;

if(!v.isEmpty())

{

for(Node<ll> \*p = v.first(); p != NULL; p = p->getnext())

{

if(!PrimeNumber(p->getElem()))

{

New.insertLast(p->getElem());

}

}

v.delList();

v = New;

cout << "\nDa xoa het so nguyen to.";

}

else cout << "\nDanh sach trong!";

system("pause");

}

ll dem[100000008];

void NumberList::DelDuplicate()

{

if(!v.isEmpty())

{

singleList<ll> New;

for(Node<ll> \*p = v.first(); p != NULL; p = p->getnext())

{

if(dem[p->getElem()]) continue;

New.insertLast(p->getElem());

dem[p->getElem()] = 1;

}

v.delList();

v = New;

cout << "\nXoa thanh cong!";

}

else cout << "\nDanh sach trong!";

system("pause");

}

void NumberList::SortInc()

{

if(!v.isEmpty())

{

v.sort();

cout << "\nSap xep thanh cong!";

}

else cout << "\nDanh sach trong!";

system("pause");

}

void NumberList::OutPut()

{

if(!v.isEmpty())

{

singleListItr<ll> itr(&v);

itr.resest();

while(itr.hasnext())

{

cout << itr.next() << " ";

}

}

else cout << "\nDanh sach trong!";

system("pause");

}..00000

IV.Phân tích thời gian chạy từng phương thức có trong các lớp

V.Danh sách tài liệu tham khảo