Name: Trần Viết Thịnh

Class: CMU-CS-316 FIS

MSSV: 27211245057

Đề bài:

Viết chương trình đọc file solieu.txt và thêm vào danh sách liên kết. Sau đó thực hiện các task sau:

- Tính tổng các nốt lẻ

- Tính trung bình cộng các số chẵn

- Đếm số phần tử lớn hơn 3 trong danh sách

- Đếm số phần tử khác nhau trong danh sách

- Hoán vị phần tử đầu và phần tử cuối trong danh sách

- IN giá trị nốt kế cuối trong danh sách

- Đảo danh sách liên kết

1.

package tree;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

public class Node {

int data;

Node next;

Node(int val) {

data = val;

next = null;

}

Node(int val, Node x) {

data = val;

next = x;

}

}

2.

package tree;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.HashSet;

import java.util.Scanner;

import java.util.Set;

import javax.swing.tree.TreeNode;

public class MyBST\_Fc2 {

// đọc file solieu.txt và thêm vào danh sách liên kết

Node head;

public MyBST\_Fc2() throws FileNotFoundException {

Scanner input = new Scanner(new File("solieu.txt"));

int n = input.nextInt();

input.nextLine();

while (input.hasNext()) {

int num = input.nextInt();

head = add(num, head);

}

}

public Node add(int val, Node first) {

if (first == null) {

return new Node(val);

} else if (first.next != null) {

first.next = add(val, first.next);

} else if (first.next == null) {

first.next = new Node(val);

}

return first;

}

void duyet(Node T) {

if (T != null) {

System.out.print(" " + T.data);

duyet(T.next);

}

}

public int TongSoLe(Node node) {

int sum = 0;

if (node == null) {

return sum;

} else {

if (node.data % 2 != 0) {

sum = TongSoLe(node.next) + 1;

} else {

sum = TongSoLe(node.next) + 0;

}

}

return sum;

}

public double TBCSoChang() {

Node node = head;

double sum = 0;

int dem = 0;

if (node == null) {

return 0.0;

}

while (node != null) {

if (node.data % 2 == 0) {

sum += node.data;

dem++;

}

node = node.next;

}

return sum / dem;

}

public String nodeKeCuoi() {

if (head == null) {

return "list rong";

}

Node prev = null;

Node curr = head;

// xác định node cuối

while (curr.next != null) {

prev = curr;

curr = curr.next;

}

return ("gia tri node ke cuoi la : " + prev.data);

}

public int count(Node node) {

int dem = 0;

if (node == null) {

return 0;

} else {

if (node.data > 3) {

dem = count(node.next) + 1;

} else {

dem = count(node.next) + 0;

}

}

return dem;

}

public void DaoLinkList() {

Node prev = null;

Node curr = head;

Node next = null;

while (curr != null) {

next = curr.next;

curr.next = prev;

prev = curr;

curr = next;

}

head = prev;

}

int soPhanTuKhacNhau() {

Set<Integer> set = new HashSet<>();

Node p = head;

while (p != null) {

set.add(p.data);

p = p.next;

}

return set.size();

}

public void hoangVi() {

if (head == null || head.next == null) {

return;

}

Node prev = null;

Node curr = head;

// xác định node cuối

while (curr.next != null) {

prev = curr;

curr = curr.next;

}

curr.next = head.next;

prev.next = head;

head.next = null;

head = curr;

}

void in() {

duyet(head);

}

public static void main(String[] args) throws FileNotFoundException {

MyBST\_Fc2 a = new MyBST\_Fc2();

a.in();

System.out.println("\n -> số các giá trị le: " + a.TongSoLe(a.head));

System.out.println(" -> Trung binh các số chẳn: " + a.TBCSoChang());

System.out.println(" -> các số lớn hơn giá trị 3 : " + a.count(a.head) + " numbers");

System.out.println(" -> So Phan Tu Khac Nhau La: " + a.soPhanTuKhacNhau());

a.hoangVi();

System.out.println(" -> List sau khi hoán vị:");

a.in();

System.out.println("\n " + a.nodeKeCuoi());

System.out.println(" -> list sau khi đảo");

a.DaoLinkList();

a.in();

}

}

