자료구조론 HW4

김정선 교수님

소프트웨어융합대학 소프트웨어학과

2017012197 여채린

package hw4;

import java.util.Stack;

import org.junit.Test;

public class DecimalToBinary {

public void deciTobin(int n) {

Stack<Integer> N = new Stack<Integer>();

while(n>0){

N.push(n%2);

n/=2;

}

if(N.isEmpty()) {

System.out.println(0);

}

while(!N.isEmpty()) {

System.out.print(N.pop());

}

}

public void deciTobinRec(int n) {

if(n == 1 ) {

System.out.print(1);

}

else if(n==0) {

System.out.print(0);

}

else {

deciTobinRec(n/2);

System.out.print(n%2);

}

}

}

package hw4;

import java.util.Stack;

import org.junit.Test;

public class Quiz {

boolean Solve(int start, int[] boxes) {

Stack<Integer> IndexStack = new Stack<Integer>();

Stack<Integer> ValueStack = new Stack<Integer>();

Stack<Boolean> DirectionStack = new Stack<Boolean>();

IndexStack.push(start);

ValueStack.push(boxes[start]);

DirectionStack.push(true);

while(!IndexStack.isEmpty() &&  ValueStack.peek() != 0) {

if(DirectionStack.peek() && CheckLength(boxes.length, IndexStack.peek()+ValueStack.peek())){

MoveMarker(boxes, IndexStack, ValueStack ,DirectionStack.peek());

DirectionStack.push(true);

}

else if(!DirectionStack.peek() && CheckLength(boxes.length, IndexStack.peek()-ValueStack.peek())){

MoveMarker(boxes, IndexStack, ValueStack ,DirectionStack.peek());

DirectionStack.push(true);

}

else if(DirectionStack.peek() && !CheckLength(boxes.length, IndexStack.peek()+ValueStack.peek())) {

DirectionStack.pop();

DirectionStack.push(false);

}

else if(!DirectionStack.peek() && !CheckLength(boxes.length, IndexStack.peek()-ValueStack.peek())) {

IndexStack.pop();

ValueStack.pop();

DirectionStack.pop();

if(!IndexStack.isEmpty()) {

DirectionStack.pop();

DirectionStack.push(false);

}

}

if(IndexStack.isEmpty() ||  ValueStack.peek() == 0)

break;

Stack<Integer> CopyIndexStack = new Stack<Integer>();

CopyIndexStack.addAll(IndexStack);

if(CheckVisted(CopyIndexStack ,IndexStack.peek())) {

IndexStack.pop();

ValueStack.pop();

DirectionStack.pop();

while(!DirectionStack.peek()) {

IndexStack.pop();

ValueStack.pop();

DirectionStack.pop();

}

DirectionStack.pop();

DirectionStack.push(false);

}

}

if(IndexStack.isEmpty()) {

return false;

}

else {

return true;

}

}

void MoveMarker(int[] boxes, Stack<Integer> IndexStack, Stack<Integer> ValueStack , boolean Direction) {

if(Direction) {

IndexStack.push(IndexStack.peek() + ValueStack.peek());

ValueStack.push(boxes[IndexStack.peek()]);

}

else {

IndexStack.push(IndexStack.peek() - ValueStack.peek());

ValueStack.push(boxes[IndexStack.peek()]);

}

}

boolean CheckLength(int Length ,int Index) {

if(0 <= Index && Index< Length) {

return true;

}

else {

return false;

}

}

boolean CheckVisted(Stack<Integer> CopyIndexStack ,int TopIndexStack) {

CopyIndexStack.pop();

while(!CopyIndexStack.isEmpty()) {

if(TopIndexStack == CopyIndexStack.pop()) {

return true;

}

}

return false;

}

void PrintStack(Stack<Integer> IndexStack) {

while(!IndexStack.isEmpty()) {

System.out.println(IndexStack.pop());

}

}

}