Lap 4

1. Đổi số nguyên dương ra dạng nhị phân (có trong bài giảng)

import java.util.\*;

class stack<T> {

private T[] a;

private int maxSize, top;

public stack(int size) {

maxSize = size;

a= (T[]) new Object[size];

top=-1;

}

public boolean isEmpty() {

return top==-1;

}

public void push(T x) {

a[++top] =x;

}

public T pop() {

return a[top--];

}

}

public class Bai1 {

public static void main(String[] args) {

stack s = new stack(100);

Scanner sc = new Scanner(System.in);

System.out.print("Nhap n: ");

int n = sc.nextInt();

while (n>0) {

s.push(n%2);

n=n/2;

}

System.out.print("Day nhi phan: ");

while (!s.isEmpty())

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

}

}

1. Đảo ngược chuổi theo từ

“anh yêu em” đảo thành “em yêu anh”

“cấm không được đứng đây” đảo thành “đây đứng được không cấm”

import java.util.\*;

public class Bai2 {

public static void main(String[] args) {

stack <String> s = new stack(100);

Scanner sc = new Scanner(System.in);

System.out.print("Nhap chuoi: ");

String st = sc.nextLine();

StringTokenizer stk = new StringTokenizer(st, " ", false);

while (stk.hasMoreTokens())

s.push(stk.nextToken());

System.out.print("Chuoi dao nguoc: ");

while (!s.isEmpty())

System.out.print(s.pop() + " ");

}

}

1. Kiểm tra biểu thức hợp lệ về mở ngoặc và đóng ngoặc

import java.util.\*;

public class Bai3 {

public static boolean bieuThucHopLeKhong(String expression) {

stack<Character> stack = new stack<>(expression.length());

for (char c : expression.toCharArray()) {

if (c == '(') {

stack.push(c);

} else if (c == ')') {

if (stack.isEmpty() || stack.pop() != '(') {

return false;

}

}

}

return stack.isEmpty();

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Nhap bieu thuc: ");

String expression = sc.nextLine();

if (bieuThucHopLeKhong(expression)) {

System.out.println("Bieu thuc hop le");

} else {

System.out.println("Bieu thuc khong hop le");

}

}

}

1. Khử bỏ đệ quy bài toán tháp Hà nội

import java.util.\*;

public class Bai4 {

public static void thapHaNoiKhongDeQuy(int n, char nguon, char tamthoi, char dich) {

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

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

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

for (int i = n; i >= 1; i--) {

stackA.push(i);

}

char temp;

if (n % 2 == 0) {

temp = tamthoi;

tamthoi = dich;

dich = temp;

}

int soBuoc = (int) Math.pow(2, n) - 1;

for (int buoc = 1; buoc <= soBuoc; buoc++) {

if (buoc % 3 == 1) {

chuyenDiaGiuaHaiCot(stackA, stackC, nguon, dich);

} else if (buoc % 3 == 2) {

chuyenDiaGiuaHaiCot(stackA, stackB, nguon, tamthoi);

} else if (buoc % 3 == 0) {

chuyenDiaGiuaHaiCot(stackB, stackC, tamthoi, dich);

}

}

}

public static void chuyenDiaGiuaHaiCot(Stack<Integer> nguon, Stack<Integer> dich, char cotNguon, char cotDich) {

if (nguon.isEmpty() && dich.isEmpty()) {

System.out.println("Thap Ha Noi rong");

return;

} else if (nguon.isEmpty()) {

int topDisk = dich.pop();

nguon.push(topDisk);

System.out.println("Chuyen dia " + topDisk + " tu " + cotDich + " den " + cotNguon);

} else if (dich.isEmpty()) {

int topDisk = nguon.pop();

dich.push(topDisk);

System.out.println("Chuyen dia " + topDisk + " tu " + cotNguon + " den " + cotDich);

} else {

int srcTop = nguon.peek();

int destTop = dich.peek();

if (srcTop > destTop) {

int topDisk = dich.pop();

nguon.push(topDisk);

System.out.println("Chuyen dia " + topDisk + " tu " + cotDich + " den " + cotNguon);

} else {

int topDisk = nguon.pop();

dich.push(topDisk);

System.out.println("Chuyen dia " + topDisk + " tu " + cotNguon + " den " + cotDich);

}

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Nhap so luong dia: ");

int n = sc.nextInt();

thapHaNoiKhongDeQuy(n, 'A', 'B', 'C');

}

}

1. Khử bỏ đệ quy Quick Sort

import java.util.\*;

public class Bai5 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

ArrayList<Integer> daySo = new ArrayList<>();

System.out.print("Nhap day so (nhap 0 de ket thuc): ");

int soNhap;

do {

soNhap=sc.nextInt();

daySo.add(soNhap);

} while (soNhap != 0);

daySo.remove(daySo.size()-1);

int[] arr = new int[daySo.size()];

for (int i=0; i<daySo.size(); i++) {

arr[i] = daySo.get(i);

}

System.out.print("Mang truoc khi sap xep: " + daySo.toString());

System.out.println(" ");

quickSort(arr);

System.out.print("Mang sau khi sap xep: " + Arrays.toString(arr));

}

public static void quickSort(int[] arr) {

if (arr == null || arr.length == 0) {

return;

}

int thap = 0;

int cao = arr.length - 1;

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

stack.push(thap);

stack.push(cao);

while (!stack.isEmpty()) {

cao = stack.pop();

thap = stack.pop();

int chiSo = partition(arr, thap, cao);

if (chiSo - 1 > thap) {

stack.push(thap);

stack.push(chiSo - 1);

}

if (chiSo + 1 < cao) {

stack.push(chiSo + 1);

stack.push(cao);

}

}

}

public static int partition(int[] arr, int thap, int cao) {

int chiSo = arr[cao];

int i = (thap - 1);

for (int j = thap; j < cao; j++) {

if (arr[j] < chiSo) {

i++;

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

int temp = arr[i + 1];

arr[i + 1] = arr[cao];

arr[cao] = temp;

return i + 1;

}

}