Name:Sudheep S Dept:CSE Reg No:22CS!71 Date:14.11.24

1.Bull and Sell Stock

import java.util.Arrays;

class Main {

static int a(int[] p, int s, int e) {

int r = 0;

for (int i = s; i < e; i++) {

for (int j = i + 1; j <= e; j++) {

if (p[j] > p[i]) {

int c = (p[j] - p[i]) + a(p, s, i - 1) + a(p, j + 1, e);

r = Math.max(r, c);

}

}

}

return r;

}

static int b(int[] p) {

return a(p, 0, p.length - 1);

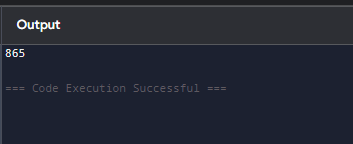
}

public static void main(String[] args) {

int[] p = {100, 180, 260, 310, 40, 535, 695};

System.out.println(b(p));

}

}  
  
  
  
Time Complexity:O(n)  
Space Complexity:O(1)  
  
2.Coin Exchange  
  
import java.io.\*;

public class Main {

static int a(int[] c, int m, int s) {

if (s == 0) return 0;

int r = Integer.MAX\_VALUE;

for (int i = 0; i < m; i++) {

if (c[i] <= s) {

int t = a(c, m, s - c[i]);

if (t != Integer.MAX\_VALUE && t + 1 < r)

r = t + 1;

}

}

return r;

}

public static void main(String[] args) {

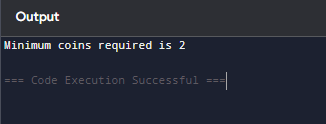
int[] c = {9, 6, 5, 1};

int m = c.length;

int s = 11;

System.out.println("Minimum coins required is " + a(c, m, s));

}

}  


Time Complexity: O(m^sum)  
Space Complexity:O(sum)

3.First and Last Occurance  
  
import java.io.\*;

class GFG {

public static void a(int[] arr, int x) {

int n = arr.length;

int f = -1, l = -1;

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

if (x != arr[i]) continue;

if (f == -1) f = i;

l = i;

}

if (f != -1) {

System.out.println( f);

System.out.println(l);

} else {

System.out.println("Not Found");

}

}

public static void main(String[] args) {

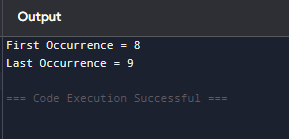
int[] arr = { 1, 2, 2, 2, 2, 3, 4, 7, 8, 8 };

int x = 8;

a(arr, x);

}

}

  
Time Complexity: O(n)  
Space Complexity:O(1)

4.Find Transition Point  
  
import java.util.\*;

class GFG {

static int a(int[] arr, int n) {

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

if (arr[i] == 1)

return i;

return -1;

}

public static void main(String[] args) {

int[] arr = {0, 0, 0, 0, 1, 1};

int n = arr.length;

int p = a(arr, n);

if (p >= 0)

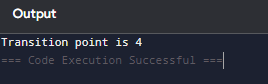
System.out.print("Transition point is " + p);

else

System.out.print("There is no transition point");

}

}



Time Complexity: O(n)  
Space Complexity:O(1)

5. First Repeating element  
  
import java.util.\*;

public class GFG {

public static int a(int[] arr, int n) {

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

for (int j = i + 1; j < n; j++) {

if (arr[i] == arr[j]) {

return i;

}

}

}

return -1;

}

public static void main(String[] args) {

int[] arr = { 10, 5, 3, 4, 3, 5, 6 };

int n = arr.length;

int idx = a(arr, n);

if (idx == -1) {

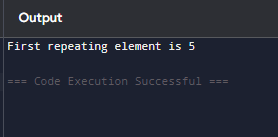
System.out.println("No repeating element found!");

} else {

System.out.println("First repeating element is " + arr[idx]);

}

}

}  
  


Time Complexity: O(n)  
Space Complexity:O(n)

6. Remove Duplicates from sorted array

import java.util.HashSet;

class GfG {

static int a(int[] arr) {

HashSet<Integer> s = new HashSet<>();

int idx = 0;

for (int i = 0; i < arr.length; i++) {

if (!s.contains(arr[i])) {

s.add(arr[i]);

arr[idx++] = arr[i];

}

}

return idx;

}

public static void main(String[] args) {

int[] arr = {1, 2, 2, 3, 4, 4, 4, 5, 5};

int sz = a(arr);

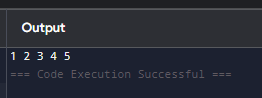
for (int i = 0; i < sz; i++) {

System.out.print(arr[i] + " ");

}

}

}

  
  
Time Complexity: O(n)  
Space Complexity:O(1)

7.Maximum Index

public class FindMaximum {

int a(int[] arr, int n) {

int m = -1;

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

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

if (arr[j] > arr[i] && m < (j - i))

m = j - i;

}

}

return m;

}

public static void main(String[] args) {

FindMaximum max = new FindMaximum();

int[] arr = { 9, 2, 3, 4, 5, 6, 7, 8, 18, 0 };

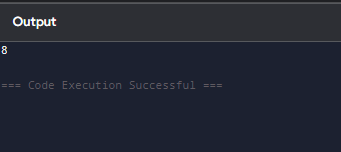
int n = arr.length;

int m = max.a(arr, n);

System.out.println(m);

}

}



Time Complexity: O(n^2)  
Space Complexity:O(1)

8.Wave Array

import java.util.\*;

public class Main {

void s(int[] arr, int a, int b) {

int t = arr[a];

arr[a] = arr[b];

arr[b] = t;

}

void w(int[] arr, int n) {

Arrays.sort(arr);

for (int i = 0; i < n - 1; i += 2)

s(arr, i, i + 1);

}

public static void main(String[] args) {

Main ob = new Main();

int[] arr = {10, 90, 49, 2, 1, 5, 23};

int n = arr.length;

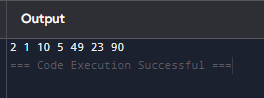
ob.w(arr, n);

for (int i : arr)

System.out.print(i + " ");

}

}



Time Complexity: O(n)  
Space Complexity:O(1)