Lab 2 – Design and Analysis of Algorithms

Tanvi Penumudy

E18CSE187

Code (Brute Force):

x = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]

y = [80,100,110,90,65,70,75,90,80,70,100,80,65,60,55,50]

l1=len(x)-1

x1=[0,0]

y1=[0,0] #temporary max

for i in range(l1):

  for j in range(i+1,l1+1): #i<j

    if((y[j]-y[i])>(y1[1]-y1[0])):

      y1[0]=y[i]

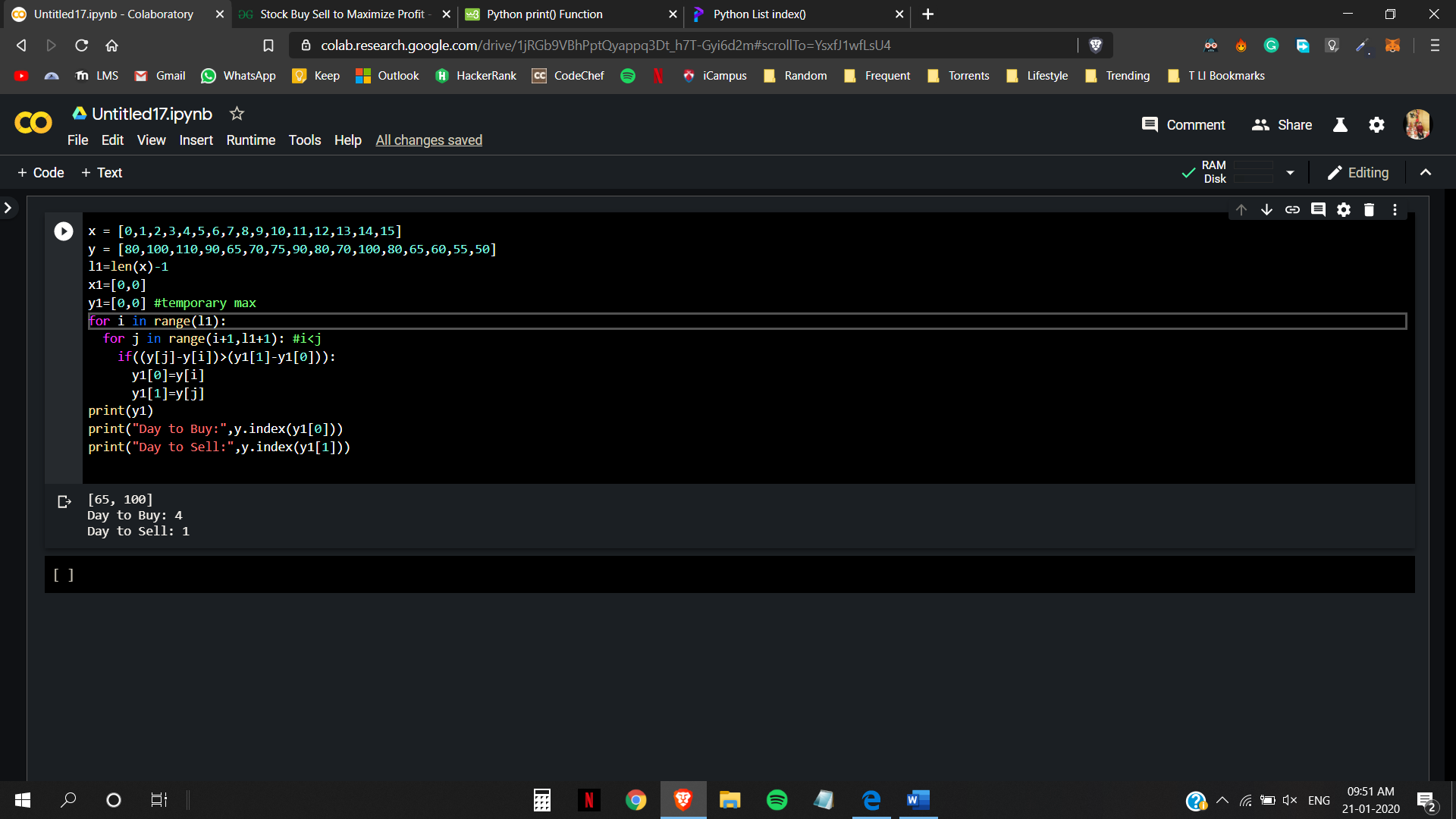
      y1[1]=y[j]

print(y1)

print("Day to Buy:",y.index(y1[0]))

print("Day to Sell:",y.index(y1[1]))

Output:



Analysis:

The time complexity of the above algorithm is n^2, since there are two ‘for loops’ one inside another making the time complexity n\*n (multiplying them), hence O(n^2).

Code:

package LabAssign2algo;

import java.util.ArrayList;

class Stockini {

int buy, sell;}

public class Lab2algo {

void lab2algo(int price[], int n)

{

if(n==1)

return;

int count=0;

ArrayList<Stockini> sol=new ArrayList<Stockini>();

int i=0;

while(i<n-1){

while((i<n-1)&&(price[i+1]<=price[i]))

i++;

if(i==n-1)

break;

Stockini e=new Stockini();

e.buy=i++;

while(i < n)&&(price[i]>=price[i - 1]))

i++;

e.sell=i-1;

sol.add(e);

count++;

}

if (count==0)

System.out.println("There is no day when buying the stock "

+ "will make profit");

else

for(int j=0;j<count;j++)

System.out.println("Buy on day no: " + sol.get(j).buy

+ " " + "Sell on day no: " + sol.get(j).sell);

return;

}

public static void main(String args[])

{

Lab2algo stock = new Lab2algo();

int y1[]={80,100,110,90,65,70,75,90,80,70,100,80,65,60,55,50};

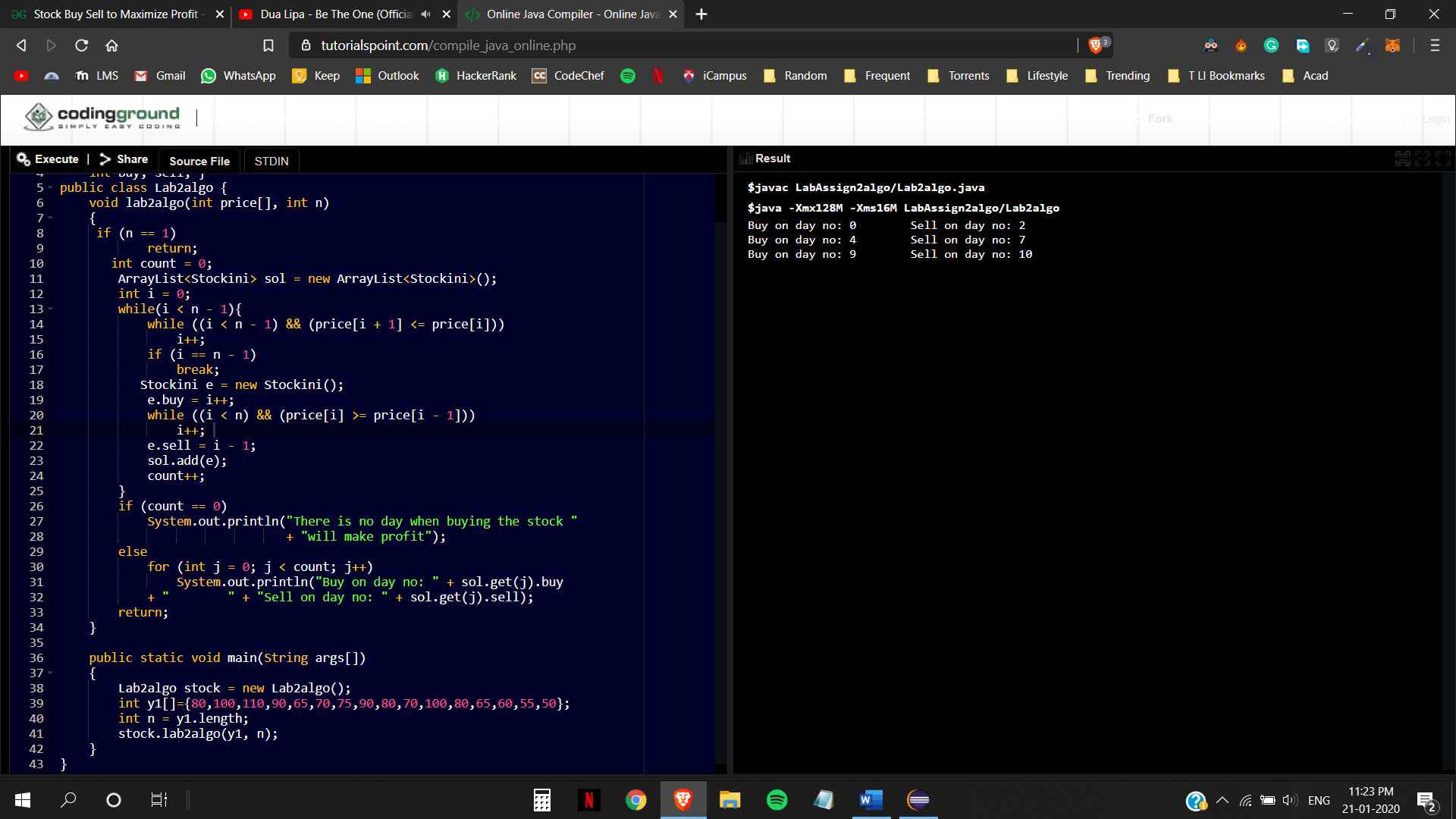
int n = y1.length;

stock.lab2algo(y1, n);

}

}

Output:



Analysis:

The resultant time complexity is in the order of n^2 as the two ‘inner while loops’ add up to give the complexity of (n-1)+n and considering the ‘outer while loop’ and the ‘for loop’ (if we take the count value as some constant k) the resultant will be (n-1)(2n-1)+k. So hence it is in the order of O(n^2).