Assignment 1: Assignment based on some simple coding problems on numbers, graphs, matrices

Name: Adnan Sadar

Class: TY IT

Roll No: 2

Batch: 3

Problem Statement 1: Find Maximum sum of any submatrix of a Matrix which is sorted row-wise and column-wise.

Code:

class Assignment1Q1 {

public static void main(String[] args) {

// Given matrix mat[][]

int[][] mat1 = { { -6, -4, -1 }, { -3, 2, 4 }, { 2, 5, 8 } };

int[][] mat2 = { { -4, -3 }, { -2, -1 } };

System.out.println(

"The array: ");

for (int i = 0; i < mat1.length; i++) {

for (int j = 0; j < mat1.length; j++) {

System.out.print(mat1[i][j]+" ");

}

System.out.println();

}

System.out.println("Output: The sum from the largest submatrix is " + maxSubMatSum(mat1));

System.out.println(

"The array: ");

for (int i = 0; i < mat2.length; i++) {

for (int j = 0; j < mat2.length; j++) {

System.out.print(mat2[i][j] + " ");

}

System.out.println();

}

System.out.println("Output: The sum from the largest submatrix is " + maxSubMatSum(mat2));

}

// Function that finds the maximum

// Sub-Matrix Sum

static int maxSubMatSum(int[][] mat) {

// Number of rows in the matrix

int n = mat.length;

// Number of columns in the matrix

int m = mat[0].length;

int i, j;

// dp[][] matrix to store the

// results of each iteration

int[][] dp = new int[n][m];

// Base Case - The largest

// element in the matrix

dp[n - 1][m - 1] = mat[n - 1][m - 1];

// To stores the final result

int res = dp[n - 1][m - 1];

// Find the max sub matrix sum for

// the last row

for (i = m - 2; i >= 0; i--) {

dp[n - 1][i] = mat[n - 1][i] + dp[n - 1][i + 1];

// Check whether the current

// sub-array yeilds maximum sum

res = Math.max(res, dp[n - 1][i]);

}

// Calculate the max sub matrix

// sum for the last column

for (i = n - 2; i >= 0; i--) {

dp[i][m - 1] = mat[i][m - 1] + dp[i + 1][m - 1];

// Check whether the current

// sub-array yeilds maximum sum

res = Math.max(res, dp[i][m - 1]);

}

// Build the dp[][] matrix from

// bottom to the top row

for (i = n - 2; i >= 0; i--) {

for (j = m - 2; j >= 0; j--) {

// Update sum at each

// cell in dp[][]

dp[i][j] = mat[i][j] + dp[i][j + 1] + dp[i + 1][j] - dp[i + 1][j + 1];

// Update the maximum sum

res = Math.max(res, dp[i][j]);

}

}

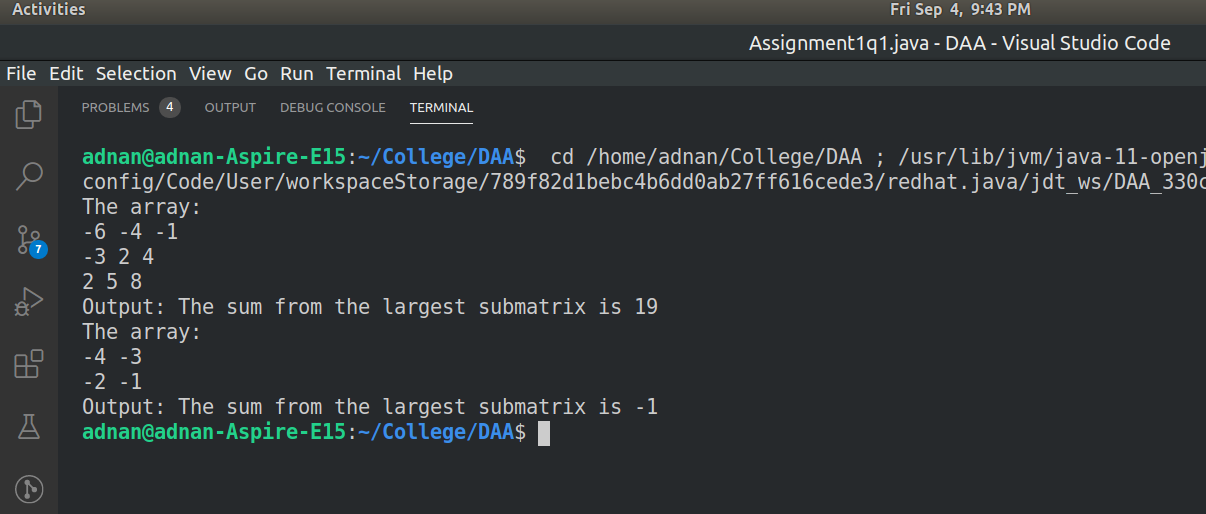
// Return the maximum sum

return res;

}

}

Output:



Problem Statement: Check whether the matrix is balanced or not

Code:

import java.util.\*;

class Assignment1q2 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int[] arr = new int[9];

int[] adjacent = { 2, 3, 2, 3, 4, 3, 2, 3, 2 };

System.out.println("Enter the 3x3 matrix: ");

for (int i = 0; i < arr.length - 1; i++) {

arr[i] = sc.nextInt();

}

Assignment1q2 ob = new Assignment1q2();

ob.checkBalanced(arr, adjacent);

}

void checkBalanced(int[] arr, int[] adjacent) {

int check = 0;

for (int i = 0; i < arr.length - 1; i++) {

if (adjacent[i] < arr[i]) {

check = 1;

break;

}

}

if (check == 1) {

System.out.println("The matrix is Unbalanced");

} else {

System.out.println("The matrix is Balanced");

}

}

}

Output:

