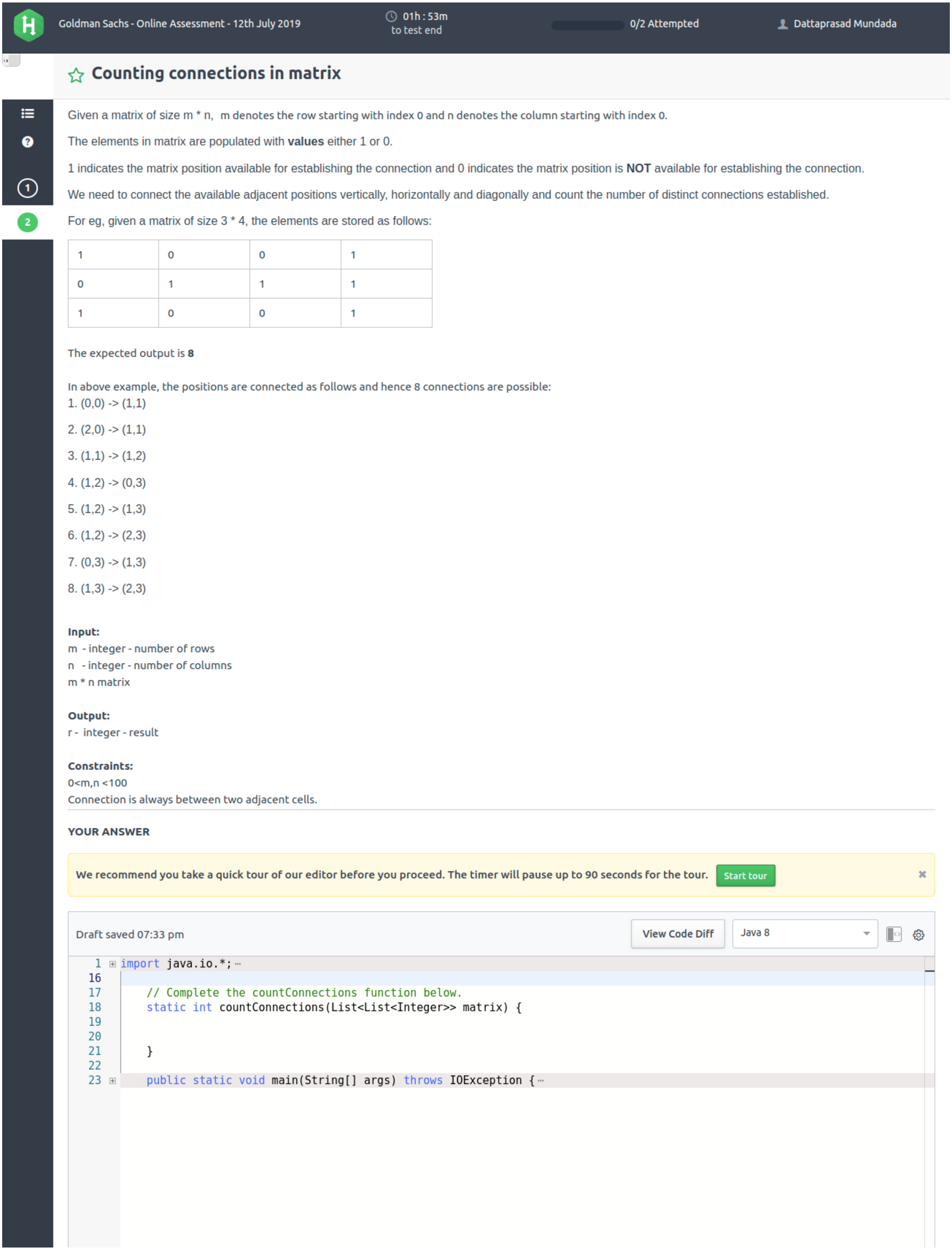


Goldman Sachs - Online Assessment - 12th July 2019

01h : 53m
to test end

0/2 Attempted

Dattaprasad Mundada

Counting connections in matrix

Given a matrix of size m * n, m denotes the row starting with index 0 and n denotes the column starting with index 0.

The elements in matrix are populated with **values** either 1 or 0.

1 indicates the matrix position available for establishing the connection and 0 indicates the matrix position is **NOT** available for establishing the connection.

We need to connect the available adjacent positions vertically, horizontally and diagonally and count the number of distinct connections established.

For eg, given a matrix of size 3 * 4, the elements are stored as follows:

1	0	0	1
0	1	1	1
1	0	0	1

The expected output is **8**

In above example, the positions are connected as follows and hence 8 connections are possible:

1. (0,0) -> (1,1)
2. (2,0) -> (1,1)
3. (1,1) -> (1,2)
4. (1,2) -> (0,3)
5. (1,2) -> (1,3)
6. (1,2) -> (2,3)
7. (0,3) -> (1,3)
8. (1,3) -> (2,3)

Input:
m - integer - number of rows
n - integer - number of columns
m * n matrix

Output:
r - integer - result

Constraints:
0<m,n <100
Connection is always between two adjacent cells.

YOUR ANSWER

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour.

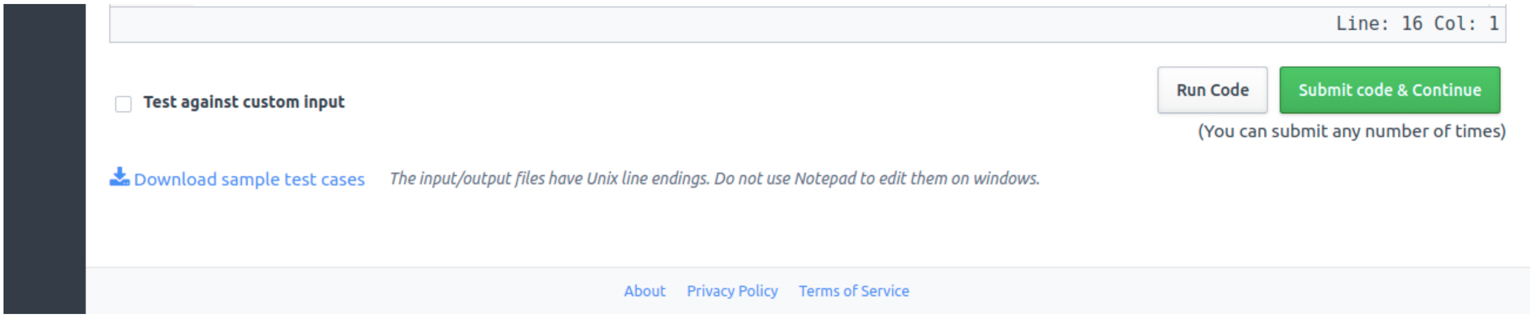
Draft saved 07:33 pm

```
import java.io.*;⋯

    // Complete the countConnections function below.
    static int countConnections(List<List<Integer>> matrix) {


    }

    public static void main(String[] args) throws IOException {⋯
```



□ Test against custom input

(You can submit any number of times)

The input/output files have Unix line endings. Do not use Notepad to edit them on windows.