UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

FACULTAD DE PRODUCCION Y SERVICIOS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

Curso: Laboratorio de Análisis y Diseño de Algoritmos

Práctica 9

Estudiante: Tapara Quispe, Fabiola Grissel

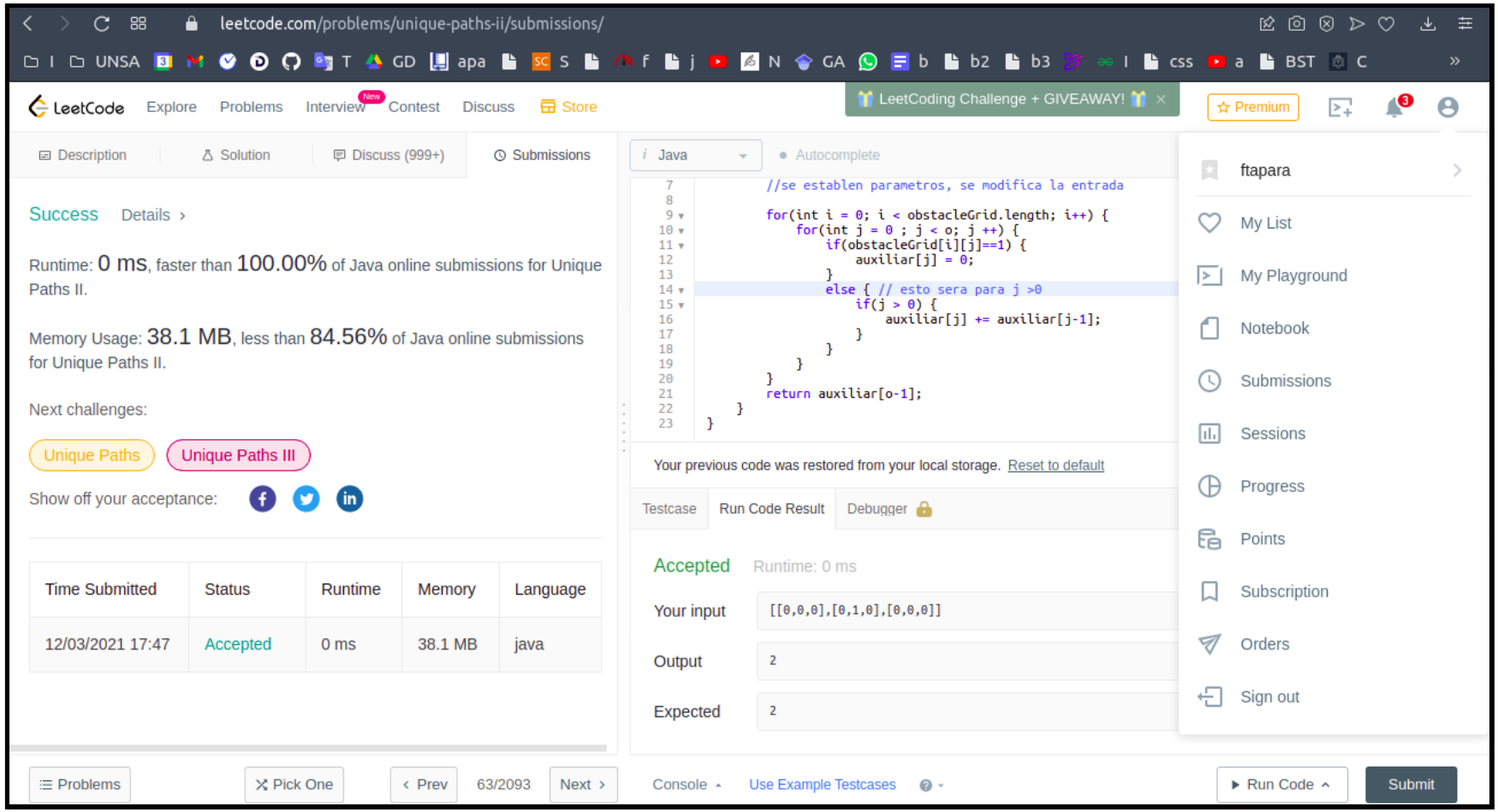
Docente: Alex Josue Florez Farfan

Sección: “B”

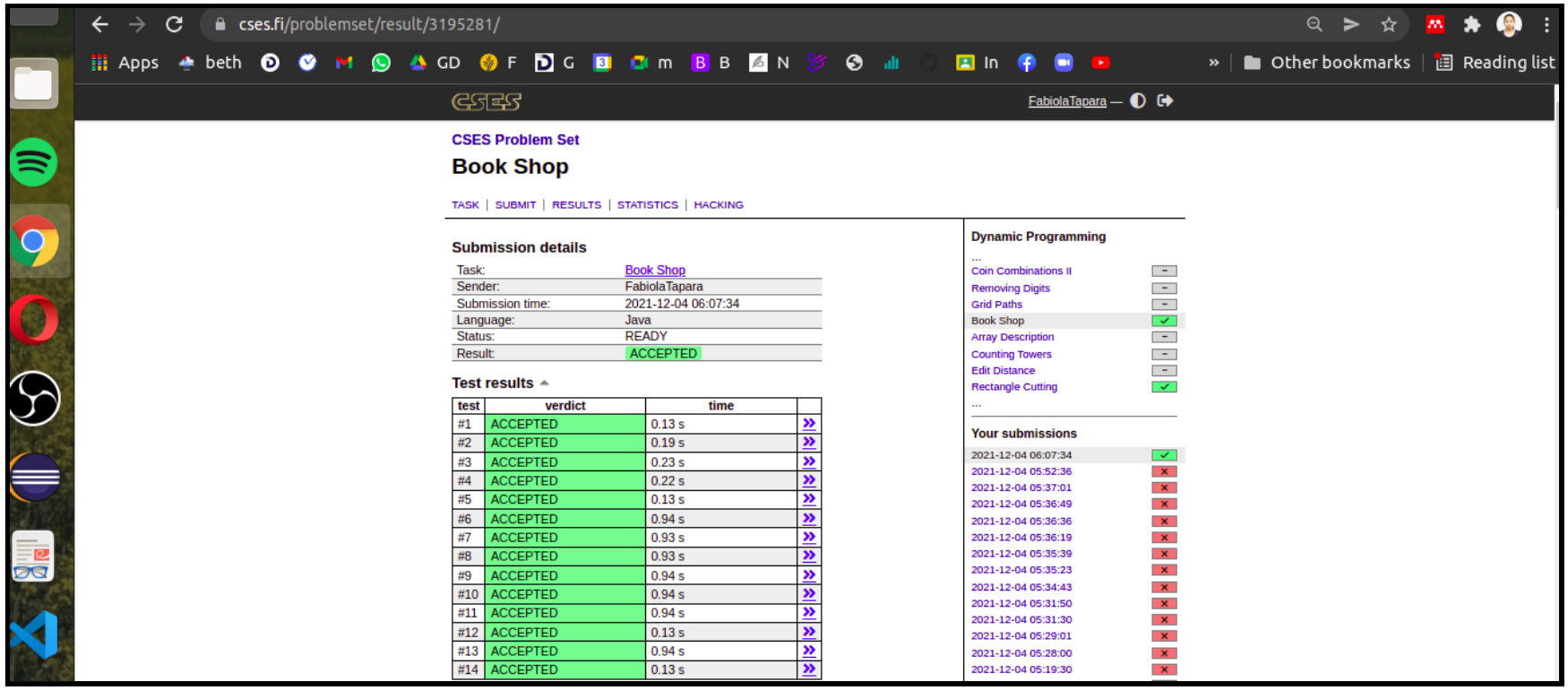
Arequipa - Perú

Diciembre 2021

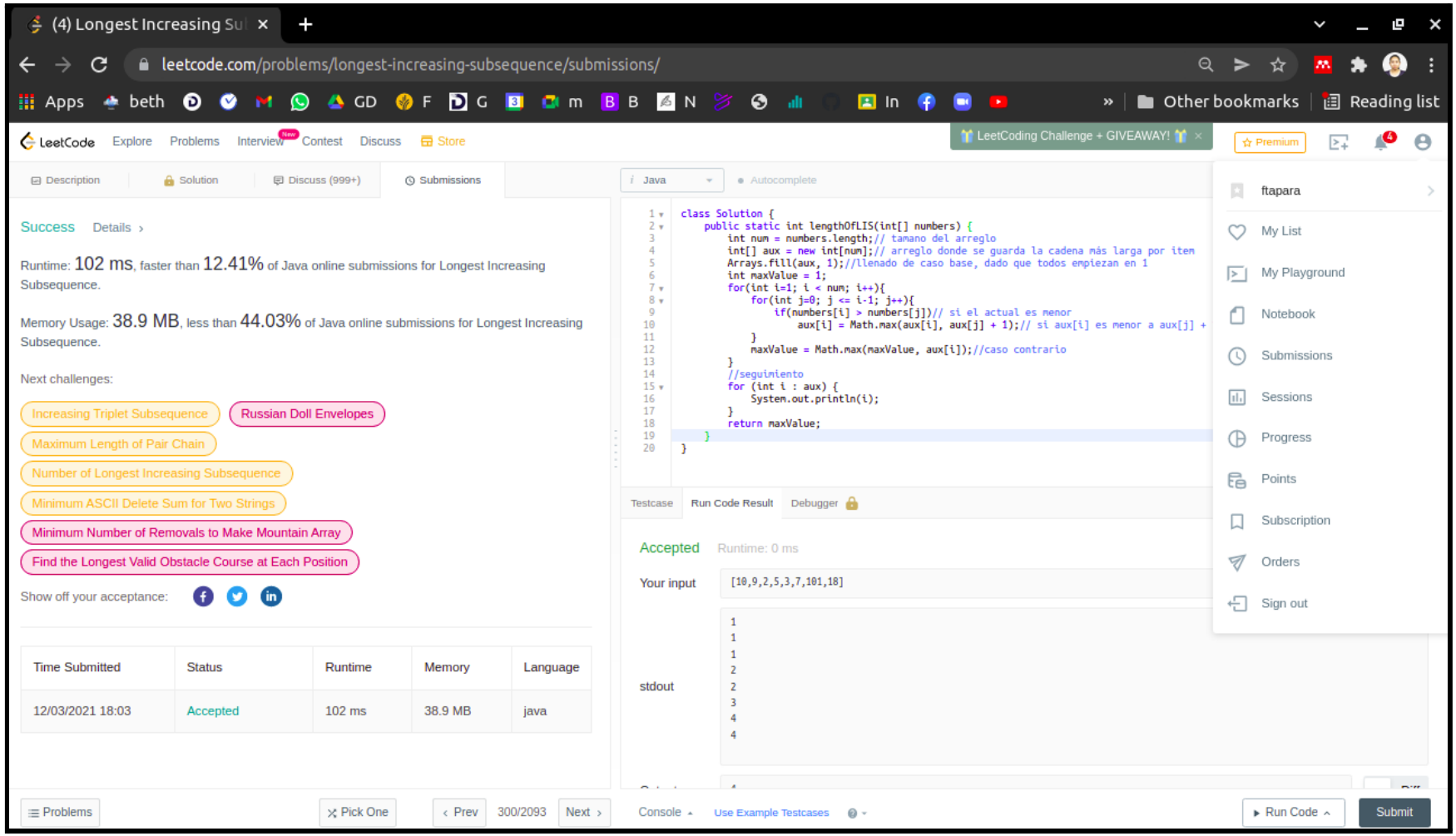
Ejercicio 1


leetcode.com/problems/unique-paths-ii/submissions/
LeetCode
Explore
Problems
Interview
Contest
Discuss
Store
LeetCoding Challenge + GIVEAWAY!
Premium
Description
Solution
Discuss (999+)
Submissions
Success
Details ›
Runtime: 0 ms, faster than 100.00% of Java online submissions for Unique Paths II.
Memory Usage: 38.1 MB, less than 84.56% of Java online submissions for Unique Paths II.
Next challenges:
Unique Paths
Unique Paths III
Show off your acceptance:
Time Submitted | Status | Runtime | Memory | Language
12/03/2021 17:47 | Accepted | 0 ms | 38.1 MB | java
Java
Autocomplete
//se establen parametros, se modifica la entrada
for(int i = 0; i < obstacleGrid.length; i++) {
for(int j = 0 ; j < o; j ++) {
if(obstacleGrid[i][j]==1) {
auxiliar[j] = 0;
}
else { // esto sera para j >0
if(j > 0) {
auxiliar[j] += auxiliar[j-1];
}
}
}
}
return auxiliar[o-1];
}
}
Your previous code was restored from your local storage. Reset to default
Testcase
Run Code Result
Debugger
Accepted
Runtime: 0 ms
Your input
[[0,0,0],[0,1,0],[0,0,0]]
Output
2
Expected
2
ftapara
My List
My Playground
Notebook
Submissions
Sessions
Progress
Points
Subscription
Orders
Sign out
Problems
Pick One
‹ Prev
63/2093
Next ›
Console
Use Example Testcases
Run Code
Submit

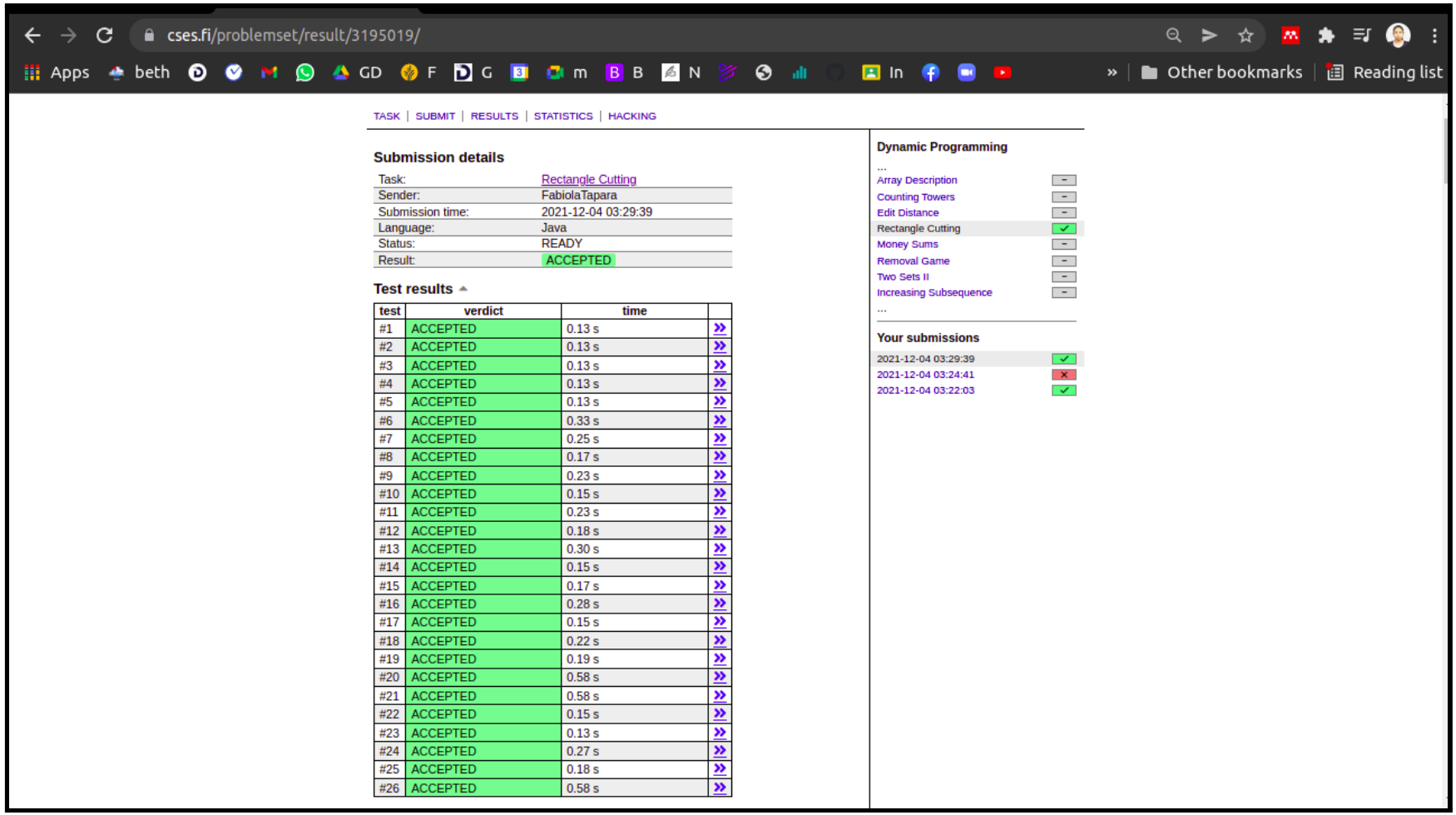
Ejercicio 2


cses.fi/problemset/result/3195281/
CSES
FabiolaTapara
CSES Problem Set
Book Shop
TASK | SUBMIT | RESULTS | STATISTICS | HACKING
Submission details
Task: Book Shop
Sender: FabiolaTapara
Submission time: 2021-12-04 06:07:34
Language: Java
Status: READY
Result: ACCEPTED
Test results
test | verdict | time
#1 ACCEPTED 0.13 s
#2 ACCEPTED 0.19 s
#3 ACCEPTED 0.23 s
#4 ACCEPTED 0.22 s
#5 ACCEPTED 0.13 s
#6 ACCEPTED 0.94 s
#7 ACCEPTED 0.93 s
#8 ACCEPTED 0.93 s
#9 ACCEPTED 0.94 s
#10 ACCEPTED 0.94 s
#11 ACCEPTED 0.94 s
#12 ACCEPTED 0.13 s
#13 ACCEPTED 0.94 s
#14 ACCEPTED 0.13 s
Dynamic Programming
Coin Combinations II
Removing Digits
Grid Paths
Book Shop
Array Description
Counting Towers
Edit Distance
Rectangle Cutting
Your submissions
2021-12-04 06:07:34
2021-12-04 05:52:36
2021-12-04 05:37:01
2021-12-04 05:36:49
2021-12-04 05:36:36
2021-12-04 05:36:19
2021-12-04 05:35:39
2021-12-04 05:35:23
2021-12-04 05:34:43
2021-12-04 05:31:50
2021-12-04 05:31:30
2021-12-04 05:29:01
2021-12-04 05:28:00
2021-12-04 05:19:30

Ejercicio 3


(4) Longest Increasing Su
leetcode.com/problems/longest-increasing-subsequence/submissions/
LeetCode
Explore
Problems
Interview
Contest
Discuss
Store
LeetCoding Challenge + GIVEAWAY!
Premium
Description
Solution
Discuss (999+)
Submissions
Success
Details
Runtime: 102 ms, faster than 12.41% of Java online submissions for Longest Increasing Subsequence.
Memory Usage: 38.9 MB, less than 44.03% of Java online submissions for Longest Increasing Subsequence.
Next challenges:
Increasing Triplet Subsequence
Russian Doll Envelopes
Maximum Length of Pair Chain
Number of Longest Increasing Subsequence
Minimum ASCII Delete Sum for Two Strings
Minimum Number of Removals to Make Mountain Array
Find the Longest Valid Obstacle Course at Each Position
Show off your acceptance:
Time Submitted
Status
Runtime
Memory
Language
12/03/2021 18:03
Accepted
102 ms
38.9 MB
java
Java
Autocomplete
class Solution {
public static int lengthOfLIS(int[] numbers) {
int num = numbers.length;// tamano del arreglo
int[] aux = new int[num];// arreglo donde se guarda la cadena más larga por item
Arrays.fill(aux, 1);//llenado de caso base, dado que todos empiezan en 1
int maxValue = 1;
for(int i=1; i < num; i++){
for(int j=0; j <= i-1; j++){
if(numbers[i] > numbers[j])// si el actual es menor
aux[i] = Math.max(aux[i], aux[j] + 1);// si aux[i] es menor a aux[j] +
}
maxValue = Math.max(maxValue, aux[i]);//caso contrario
}
//seguimiento
for (int i : aux) {
Systen.out.println(i);
}
return maxValue;
}
}
Testcase
Run Code Result
Debugger
Accepted
Runtime: 0 ms
Your input
[10,9,2,5,3,7,101,18]
stdout
1
1
1
2
2
3
4
4
ftapara
My List
My Playground
Notebook
Submissions
Sessions
Progress
Points
Subscription
Orders
Sign out
Problems
Pick One
Prev
300/2093
Next
Console
Use Example Testcases
Run Code
Submit

Ejercicio 4



cses.fi/problemset/result/3195019/

TASK | SUBMIT | RESULTS | STATISTICS | HACKING

Submission details

Task:	Rectangle Cutting
Sender:	FabiolaTapara
Submission time:	2021-12-04 03:29:39
Language:	Java
Status:	READY
Result:	ACCEPTED

Test results

test	verdict	time
#1	ACCEPTED	0.13 s
#2	ACCEPTED	0.13 s
#3	ACCEPTED	0.13 s
#4	ACCEPTED	0.13 s
#5	ACCEPTED	0.13 s
#6	ACCEPTED	0.33 s
#7	ACCEPTED	0.25 s
#8	ACCEPTED	0.17 s
#9	ACCEPTED	0.23 s
#10	ACCEPTED	0.15 s
#11	ACCEPTED	0.23 s
#12	ACCEPTED	0.18 s
#13	ACCEPTED	0.30 s
#14	ACCEPTED	0.15 s
#15	ACCEPTED	0.17 s
#16	ACCEPTED	0.28 s
#17	ACCEPTED	0.15 s
#18	ACCEPTED	0.22 s
#19	ACCEPTED	0.19 s
#20	ACCEPTED	0.58 s
#21	ACCEPTED	0.58 s
#22	ACCEPTED	0.15 s
#23	ACCEPTED	0.13 s
#24	ACCEPTED	0.27 s
#25	ACCEPTED	0.18 s
#26	ACCEPTED	0.58 s

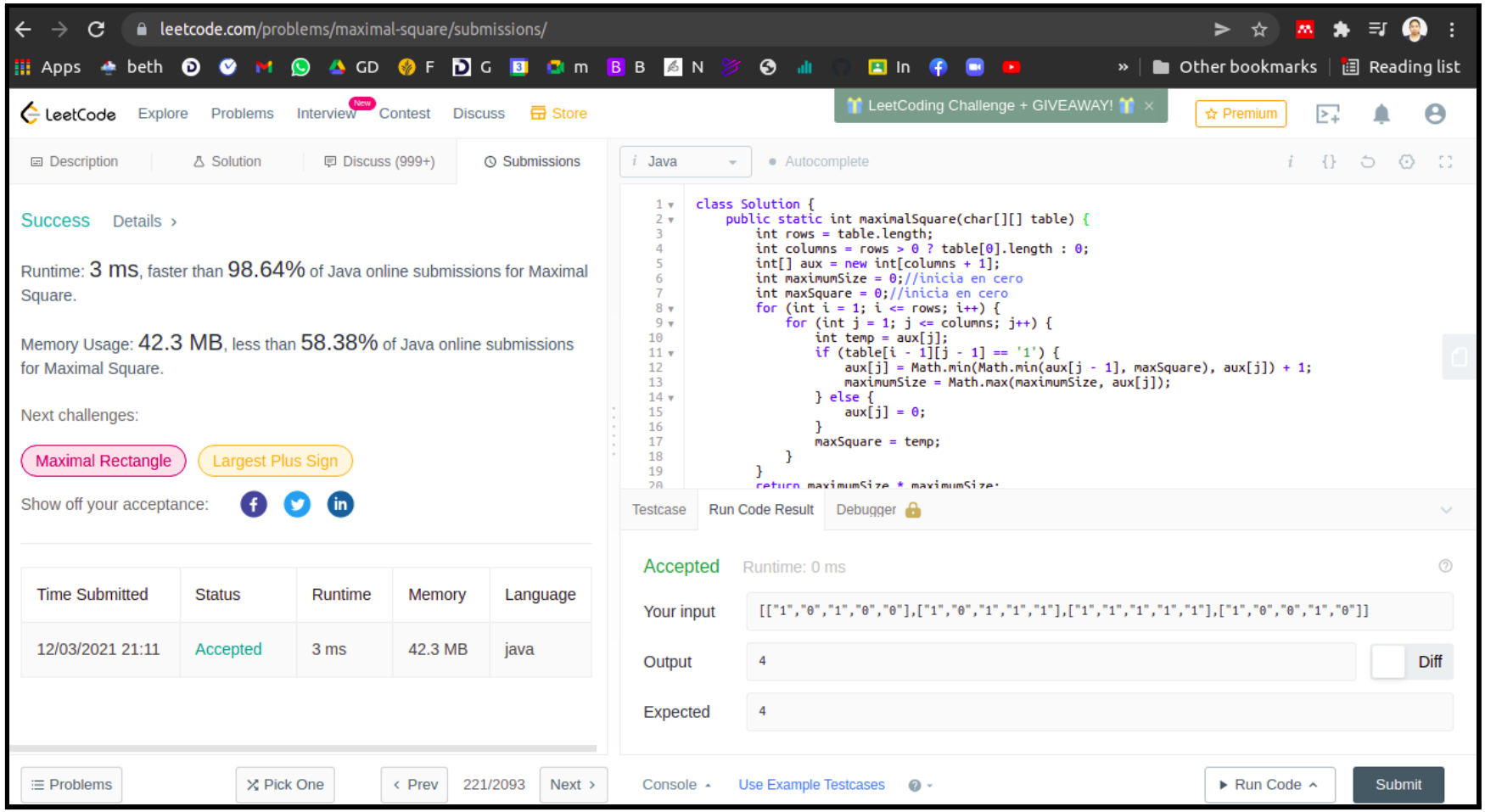
Dynamic Programming

...
Array Description
Counting Towers
Edit Distance
Rectangle Cutting
Money Sums
Removal Game
Two Sets II
Increasing Subsequence
...

Your submissions

2021-12-04 03:29:39
2021-12-04 03:24:41
2021-12-04 03:22:03

Ejercicio 5


leetcode.com/problems/maximal-square/submissions/
LeetCode
Explore
Problems
Interview
Contest
Discuss
Store
LeetCoding Challenge + GIVEAWAY!
Premium
Description
Solution
Discuss (999+)
Submissions
Success
Details ›
Runtime: 3 ms, faster than 98.64% of Java online submissions for Maximal Square.
Memory Usage: 42.3 MB, less than 58.38% of Java online submissions for Maximal Square.
Next challenges:
Maximal Rectangle
Largest Plus Sign
Show off your acceptance:
Time Submitted
Status
Runtime
Memory
Language
12/03/2021 21:11
Accepted
3 ms
42.3 MB
java
Java
Autocomplete
class Solution {
public static int maximalSquare(char[][] table) {
int rows = table.length;
int columns = rows > 0 ? table[0].length : 0;
int[] aux = new int[columns + 1];
int maximumSize = 0;//inicia en cero
int maxSquare = 0;//inicia en cero
for (int i = 1; i <= rows; i++) {
for (int j = 1; j <= columns; j++) {
int temp = aux[j];
if (table[i - 1][j - 1] == '1') {
aux[j] = Math.min(Math.min(aux[j - 1], maxSquare), aux[j]) + 1;
maximumSize = Math.max(maximumSize, aux[j]);
} else {
aux[j] = 0;
}
maxSquare = temp;
}
}
Testcase
Run Code Result
Debugger
Accepted
Runtime: 0 ms
Your input
[["1","0","1","0","0"],["1","0","1","1","1"],["1","1","1","1","1"],["1","0","0","1","0"]]
Output
4
Diff
Expected
4
Problems
Pick One
Prev
221/2093
Next
Console
Use Example Testcases
Run Code
Submit