

INTERNSHIP PROGRAM

Challenge 01 Computer Science



Challenge 01 – Computer Science

• Objectives

- Put to test the knowledge acquired during the Internship training process.
- Strengthen your development and problem-solving skills

• Challenge Details

As an algorithm engineer, you need to implement a path detection algorithm for automated harvesting. You have under your command a fleet of coffee-harvesting robots that will go uphill from the robot station to the coffee plantation, which is allocated on the side of a mountain, as is typical in Colombia, to collect ripe coffee berries. An algorithm has processed remote sensing data and determined which trees are ready to be harvested, with the following conventions:

1. If the tree is ready for harvest: A positive integer, increasing with elevation respect to the robot station.
2. If the tree is not ready for harvest: An integer negative one.

Example of the data distributed in a map:

5	5			
-1	10	5	6	4
3	8	7	4	-1
4	7	6	3	4
2	9	5	2	5
-1	5	1	2	3

The first row determines the length and width of the mountain's map (5 x 5). The robots can only visit one tree at a time and move south, east, west, and north but **only if the next tree is in a higher position with respect to the current**. They can also start harvesting at any position on the mountain, but you need to provide them with the **longest possible path uphill**

Internship Program – Development Discipline

for each one to harvest before they travel back to the station with the produce, the length is measured by the number of trees visited per path.

If there are more than one path with the same length, the most desired path would be the one with the steepest vertical distance:

Steepest distance:

5	5			
-1	10	5	6	4
3	8	7	4	-1
4	7	6	3	4
2	9	5	2	5
-1	5	1	2	3

Same length but lower vertical distance:

5	5			
-1	10	5	6	4
3	8	7	4	-1
4	7	6	3	4
2	9	5	2	5
-1	5	1	2	3

• Bonus

- Implement an optimal solution to this problem (consider time and space complexity)
- Would you tackle that issue with a different approach? Let us know 🤔

• Deliverables

- A Java console application, showing the results with the following format:
 - Input: read an external file named map.txt (it will be provided to you)
 - Output:
 - Steepest path length: <path_length>
 - List of paths:
 - <path-1>
 - <path-2>
 - ...
 - <path-n>
 - (paths should be strings of hyphen-separated elevations, for example: 12-10-9-3-1)
- Upload your code to a Git repository to track your progress. **You must send the repository link to Luis Montaña before Friday, August 5 at 1:00 pm**
- A presentation in **English**, explaining the time and space complexity of your algorithm and describing your development process. **Below, you can see the duration of the demo according to your discipline**

Discipline	Duration in Minutes
Development	10
AM	6
Testing	6

• Challenge Review

The challenge will be evaluated by a group of judges considering the following points

Item to Review	Score
Quality of Implementation	30
Output Correctness	25
Git Repository	15
Presentation	30
Optimal solution	15
Alternate solution	5
Additional Comments	
This table shows the maximum obtainable grade per item. The maximum possible score is 100/100. Optimal solution and Alternate solution items correspond to the Bonus category and will be added to the total score up to a maximum total score of 100/100.	

• Contacts

If you have any questions, do not hesitate to contact us

Diego Lopez: diegoivan.lopez@endava.com

Luis Montaña: luis.montano@endava.com

Carlos Reyes: carlosandres.reyes@endava.com

Juan Carlos Suarez: juancarlos.suarezjaimenes@endava.com