Assignment Analysis 1, 2021/2022, Period 1

Orbital survey

summative assignment

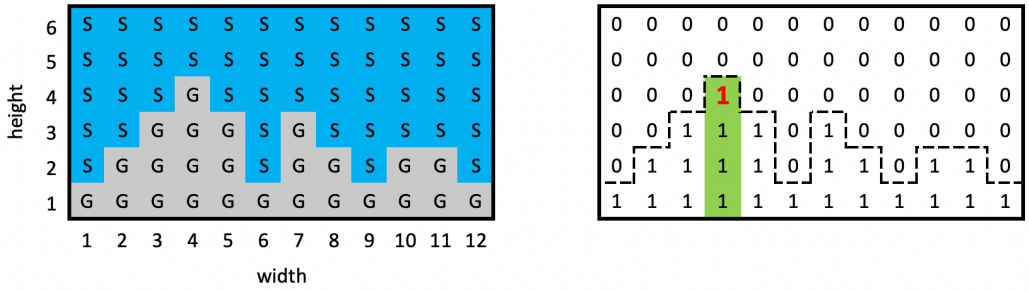
Assignment objectives

- Evaluate the ability to apply concepts learned in Analysis 1 (numeral systems, Boolean logic, sets, algorithms) to problem solving.

Problem description

An automated probe is sent to conduct orbital surveys of nearby planets. The scientists are particularly interested in the shape and the configuration of the planets' surfaces. The probe flies over a surface and scans the local region. Each scan is a 2D image, which, in the raw format, is temporarily saved as a binary sequence of 1's and 0's. 1's represent solid ground (mountains), while 0's represent open sky.

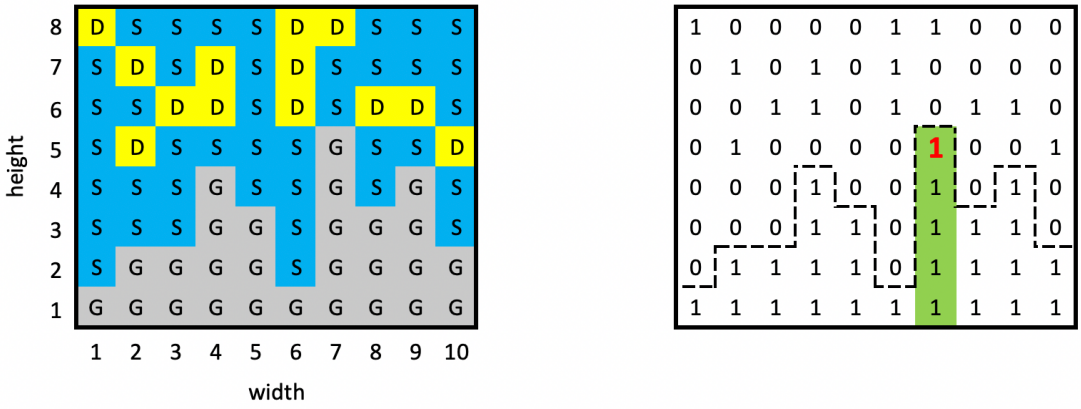
For example, a 12 x 6 area given in the image below (left), will be stored as the binary sequence in the right. Note, dashed lines and colors are there to help you visualize the problem, and are not the part of the encoded data.



Your task is to write a program that will analyze the binary sequence and use it to determine the maximum height of the scanned area. In the example above, the answer is 4 (see the column in green).

However, there could be metallic dust above the ground, interfering with the probe's sensors. If dust is present, it will generate a false-positive and the binary value of that particular cell will be erroneously saved as 1, too. The scientists have looked at the problem and determined, that the interference **never occurs exactly on top of any peaks**, i.e. there will be always at least one 0 between the peak and the metallic dust. Therefore, the algorithm should be able to filter out the noise.

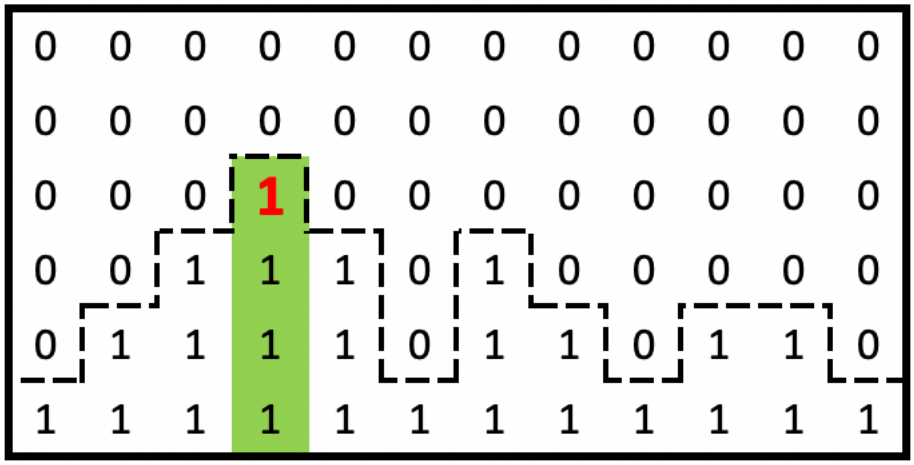
For example, below is the 10 x 8 image of an area with the noise generated by metallic dust (cells marked as D). Please note that both G cells and D cells register as 1s. As stated in the paragraph above, there is assurance that a D cell can never be on top of a G cell. For this particular case, the highest peak measurement, after filtering the noise, should be 5.



Data format

The probe scans the area from left to right, row by row, starting at the top-left cell, and ending with the bottom right cell. Then, 2D image is encoded in 1D array (list) in the same order as the cells have been scanned.

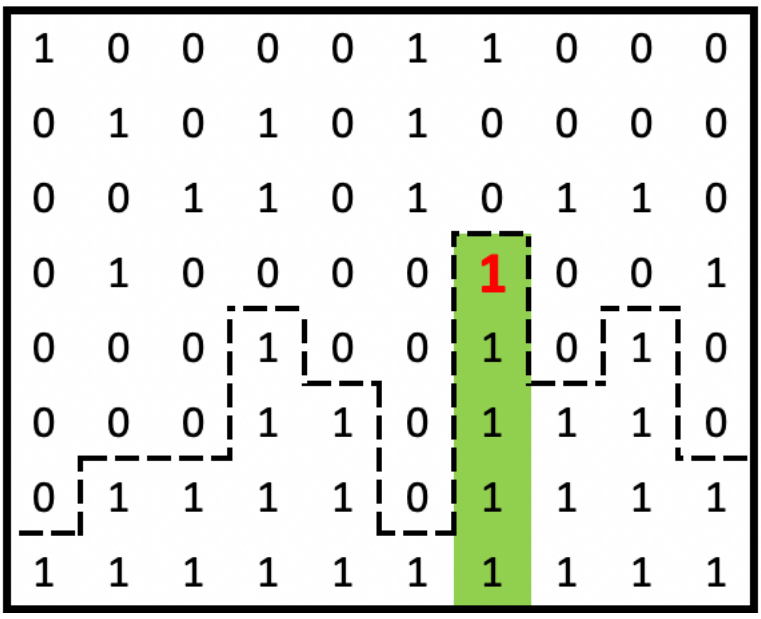
For example, the first 12 x 6 image:



as 1D binary list will look like:

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

The second 10 x 8 image:



as 1D binary list will look like:

[1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

The probe then sends the data back to Earth, consisting of **three integer values**. The **first** value is the **number of rows** in the image (height). The **second** value is the **number of columns** in the image (width). Finally, the **binary sequence is converted to its decimal equivalent**, and **that number is sent as the third value**.

Important reminder: when converting a binary sequence to a decimal value, leading zeroes will be discarded. For example: $0010_2 \rightarrow 2_{10} \rightarrow 10_2$, $00001001_2 \rightarrow 9_{10} \rightarrow 1001_2$.

Problem statement

Read three numeral inputs from the standard input – the user will input one number at a time. **Do not print any text to the user**. For the given values, find the maximal height of the terrain, according to the process and the format described above. Then, print that number as an integer value. Again, **please do not add any additional text**.

Note: sequence $000000000000000000000000000010000000000111010000001111011011011111111111_2$ (the first example) is 17607763390463_{10}.

Also,
100001100001010100000011010110010000100100010010100000110111001111011111111111111$_2$ (the second example) is 6331847315641105776476615$_{10}$.

Example input and output:
Input is given in BLUE. Required output is given in RED.

Case 1:
6
12
17607763390463
4

Case 2:
8
10
633184731564110577647615
5

Case 3:
8
8
2887099909389580922
6

Case 4:
2
2
7
2

Case 5:
2
2
12
0

Deliverables and working in groups

- **Python file** with source code, containing:
  - Student number and full name of all group members as comments
  - Python code that solves the problem

The submission will be made via MS Forms embedded inside of MS Team for Analysis 1. As MS Forms do not support submission of Python code, **please change .py extension to .dot before submitting the file**.
Students are required to submit the source code of their solution implemented in Python. Solution must be in a single file. No external libraries can be used (i.e. it must work on any computer with standard Python 3 installation).
The first line of the code should be a comment, containing student's id, student's first name and student's last name. If working in a group, other group members must be listed in the same manner in lines 2 and 3.

The assignment can be completed by a student alone, or in a group of maximum three students. The group can be made only from students that have the same Analysis teacher or teacher pair.
If a student is working alone, the student should submit her / his deliverable following the steps in the MS Form. The source file must be saved as studentnumber.py. Only then, the extension .py has to be changed to .dot to enable MS Forms to accept it.

If working in a group, only one group member ("lead") submits the assignment. All group members must be listed in the comments of the source file. When saving the source code, it must be named as: studentnumber1_studentnumber2_studentnumber3.py. Again, after this, the extension should be changed to .dot, before submitting it via Forms. Only the lead will submit the file via Forms, but must also choose the option of working in group, and correctly fill in student number(s) and name(s) of other team members. Each member will receive an automatic confirmation that the file has been received.

The teacher will give her / his feedback to all the students or just the one that submitted the assignment ("lead"). In the second case, that student is required to communicate the feedback to the rest of the team ("partner(s)").

Incomplete and improper submissions will be rejected!

Grading

The assignment will be evaluated as either PASS or FAIL. To get points for Analysis 1 course, student must pass the summative assignment together with passing the exam.
Students will receive feedback from the teacher via automated feedback e-mail message or directly via Microsoft Teams chat.

In order to PASS the assignment must:

- **Comply with the format**: Have input and output formatted exactly according to the instructions given above. Printing text in a wrong line will result in the assignment being rejected. This also includes printing any extra strings like “Please enter input” or “The result is:”, or values used in the code for testing.
- **Produce correct answer**: For every test case (test cases are not shared) the algorithm must give the correct answer. If one of the answers is incorrect, the assignment will be rejected.

Additionally, all members must fully understand every line of the submitted code and be able to defend it in front of a teacher if asked. If any member fails to defend the assignment, the assignment in its entirety will be evaluated as FAIL, impacting all members.

Deadlines

The students are required to submit the assignment before the midnight of **November 7th (Sunday)**. Submissions after the deadline will be rejected.
Students will receive the feedback from their teacher no later than **November 30th**.