Challenge 2: GRAPH DEGREE SEQUENCE

1 Requirements

1.1 Theory Part

1. State the definition of a graph's degree sequence.
2. Propose an algorithm to verify whether a non-increasing array of integers can be the degree sequence of a simple graph (Hint: The Erdős-Gallai theorem).
3. Given a degree sequence of a simple graph. Propose an algorithm to construct a simple graph with that given degree sequence (Hint: Havel-Hakimi Algorithm).

1.2 Programming Part

In this part, students are required to build an executable command-line argument program to determine whether a simple graph G can be constructed from a given non-negative array of integers.

- **Command-line:** *a.exe Input.dat Output.res*
- **Input File** a text file contains a list of non-negative integers $n_1, n_2, ..., n_k$. Each number is separated by at least one space.
- **Output File** a text file contains the result of your program which meets the following requirements:
  - `IMPOSSIBLE` - The given sequence can not construct a simple graph.
  - The first line is the value of n, denoting number of vertices of the constructed simple graph. The next lines contain the adjacency matrix of a possible simple graph when the given sequence can construct a simple graph.

Examples:

Input.dat	**Output.res**
0 1	IMPOSSIBLE
1 3 1 1	4 0 1 1 1 1 0 0 0 1 0 0 0 1 0 0 0

Explaination:

- In the first example, the sequence `0 1` can not construct a simple graph. Therefore, the result in **Output.res** is `IMPOSSIBLE`.
- In the second example, the sequence `1 3 1 1` is sorted into the non-increasing sequence, and this result can construct a graph in Figure 1.

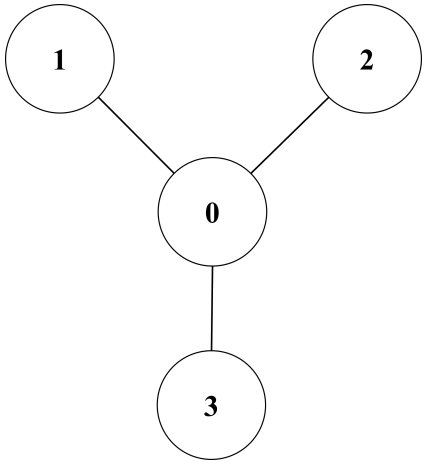

Figure 1: The constructed simple graph from the degree sequence in the second example

Based on Figure 1, we can easily identify the adjacency matrix.

2 Regulations and Evaluations

2.1 Regulations

- This challenge requires a group of 3-4 students. There **should not** be any 2 members from this challenge's group being in the same group from challenge 1.
- Only 10 first submissions is accepted.
- The submission file must be in the following format:

  [StudentID1-StudentID2-StudentID3-StudentID4.rar/.zip]

  Example:

  – Given the student codes: *21120666 - 23120888 - 23120991 - 21120999*.

  → **The name of submission file is**:

  21120666-21120999-23120888-23120991.zip/rar.

  This folder contains:

  – The report file must be presented as a document [**report.pdf**] or as a slideshow [**report.pptx**]. This file presented research answers from 1.1 and the solution of problems from 1.2.

  * If your submission is a slideshow, there must be explanation in the *Note* part of each slide.
  * Information (Name, Student's ID) must be provided on the first page (or first slide) of your report.
  * The report file should be structured, logical, clear, coherent, and answer directly to the question. The length of the submission should not exceed 15 pages for the document file, and 30 pages for the presentation slide.

  – The source code must follow the requirements in Section 1.2. The main program [**main.cpp**] should be clear, logical and commented.

2.2 Evaluation

- File submission: 100 points max where report is 70 points and source code is 30 points.
- Submission with wrong regulation will result in a "0" (zero).
- Plagiarism and Cheating will result in a "0" (zero) for the entire course and will be subject to appropriate referral to the Management Board of the the program for further action.

END