Information Theory
Assignment 3: Fast Fourier Transform

Deadline: 08.11.19 23:59
Output: File with the code has to be uploaded to Moodle, input and output files are NOT needed. Name of the single source code has to be *NameSurname.py* (For example, *IvanIvanov.py*). No other symbols allowed
Programming language: Python 3.7
Requests:

- The program must work, the code should be readable, well-structured and should contain English comments
- NO extension of a deadline. Works sent after the deadline will NOT be evaluated
- Assignment is strictly individual
- Deviations from assignment requirements will lead to *penalties* or even *annulation* of the assignment grade
- You can only import *matplotlib* explicitly. You can then use other libraries which *mathplotlib* imports transitively, even numpy, but only for image reading and writing and general-purpose matrix operations. It is explicitly forbidden to use any library functions implementing FFT
- We will be using MOSS (Measure of Software Similarity) as a test for plagiarism. Be reminded that a score of 0 will be assigned to any submissions suspected of plagiarism pending a full investigation as per IU policies.

Evaluation criteria:
0 (0%) - no submission or late submission
1 (20%) - required functionality is not achieved
2 (40%) - required functionality is lower than 50%
3 (60%) - required functionality is 50-80%
4 (80%) - required functionality is 80-100%
5 (100%) - well-structured readable correct code with English comments
Task:
Write a program that ‘compress’ and ‘decompress’ target files with grayscale image using Fast Fourier Transform. So you need to implement 2-dimensional FFT and Inverse FFT. The ‘compress’ means perform FFT and remove some resulting values by the threshold (the value of the threshold is up to you, but you should keep not more than 70% of the quality). The ‘decompress’ means perform InverseFFT under result of the FFT and cut-off. For more details (as an implementation details see the reference book).

Inputs:

Your program has to read all input files from directory *“inputs/”*, which will be in the root with your source file. Each input file is a grayscale picture with extension *“*.tif”*. Each input file name can contain only English letters and/or numbers. For example, *“Original35.tif”*, *“file.tif”*, *“555IMAGE555.tif”*, etc. Do not modify input files, they will be in read-only mode. Consider that *“inputs/”* directory may be empty or contain at most 100 files. The size of each picture should be 2^n*2^k pixels, where *n* and *k* are natural numbers greater than 0. Pictures can have different sizes.

Outputs:

In the same directory with your source code after reading the inputs your program should create a directory *“NameSurnameOutputs/”* (For example, *IvanIvanovOutputs/*), which should contain one resulting files for each file from *“inputs/”.* (For example, for *original.tif* file output should be called *originalCompressed.tif*)