Department of Electronics, Telecommunications and Informatics, University of Aveiro

Algorithms and Data Structures

2023/2024 - 1st Semester

1st Work - The TAD image8bit

Deadline for submission: 24 November 2023

This work has two main objectives:

1. Developing and testing the **image8bit** TAD, which allows images of grey levels to be instantiated and operated on, where each *pixel* can take an intensity value between 0 and 255 (8 bits).

2. Analyse the computational complexity of the `ImageLocateSubImage(...)function,` which determines the location of a sub-image in a given image, if any, and the `ImageBlur(...)function,` which applies a filter to an image and makes it dull.

1 - Developing the image8bit TAD

Based on the supplied **image8bit.h** and **image8bit.c** files, you should:

1. **Analyse** both files carefully.
2. **Complete the development of** the functions specified in the interface file but not yet finalised in the implementation file.
3. **Test** all the functions developed using the test programmes provided.

Attention:

- The **image8bit.h** interface file specifies the different functions of the TAD and **must not be altered**.

- For each function, the associated preconditions, postconditions and error situations are indicated, which must be taken into account.

- If it makes code development easier, additional auxiliary functions (**static**) can be defined and implemented in the **image8bit.c** file.

- The **code** developed should be **clear** and appropriately **commented**: the identifiers chosen for the variables and the structure of the code, as well as any comments, should be sufficient for understanding.

- There is no need to submit a report on the development of the TAD.

- Only the **image8bit.c** file - with the identification of its author(s) - should be submitted.

2 - Analysing the complexity of the `ImageLocateSubImage()` function

After completing the development of the TAD, it is necessary to analyse the computational efficiency of the algorithm developed for the `ImageLocateSubImage()` function.

To do this you must:

1. Carry out a sequence of tests, with images of different sizes, recording and analysing the number of **comparisons** made involving the grey value (i.e. the intensity) of the *pixels in* the images.

2. Carry out a **formal analysis** of the algorithm's complexity, for the **best case** and the worst **case.**
   worst case scenario.

3. Compare the results obtained in the two previous tasks.

3 - Analysing the complexity of the `ImageBlur()` function

To do this you must:

1. Carry out a sequence of tests with images and filters of different sizes and analyse the evolution of the number of operations you consider relevant.

2. Carry out a **formal analysis** of the algorithm's complexity.

Note: Although probably not the most immediate implementation, it is possible to implement this function with an algorithm that is proportional to the number of pixels in the image to be processed, without depending on the size of the window used to blur the image.

4 - Write a short report (max. 5 pages). The report should include:

For each of the functions analysed:

i) A table with the results of the tests carried out.
ii) Formal analysis of the complexity of the functions in question.

For the `ImageBlur()` function, the comparison of different algorithmic strategies for solving the problem will be valued.

5 - Evaluation Criteria

- **Development and testing of the requested functions (40%)**
  - The quality and clarity of the code and comments will be assessed
  - Any memory leaks will be checked for

- **Report: (50%)**
  - General aspects/Presentation/Conclusion
  - Analysing the complexity of the `ImageLocateSubImage()` function
    - Experimental data
    - Formal Analysis

- Analysing the complexity of the `ImageBlur()` function
  - Experimental data
  - Formal Analysis
  - Comparative Analysis Basic Algorithm/Improved Algorithm

- **Collaboration in Peer Review (2 papers) (10%)**
  - Test the operation of the code and assess its quality and clarity

The **final mark for the assignment** will be the weighted average of the teacher's mark and that of the classmates (each assignment will be assessed by 3-4 different students):

Final Grade = 75% Teacher Grade + 25% Average Peer Grade

Peer evaluation is subject to validation by the lecturers and may be discarded if it is found to be a manifestly incorrect assessment of the work submitted.

Attention:

- The work should be done in groups of 2 students.
- An online platform will be made available where they can check the correct functioning of the functions as they are developed.
- The work (**image8bit.c file** + **report**) will be delivered via the eLearning platform.
- After submitting their work, each student will receive information about the two pieces of work they have to revise. This task is an integral part of the assignment.