MATLAB Programming NYCU Fall 2024 LAB#11 2024/12/12

Part A: Main Task (C/C++ MEX function):

In this lab, you will code a function in C, compile it into a MEX file, and then use it within MATLAB. The function you will implement has the following functionalities:

- The first input **A** is a **double** array (any dimensions).
- The second input **B** is a **double** array with 2 columns. Each row in **B** represents a half-open interval. The intervals should all be non-overlapping.
- The output **C** is an integer array (use **int32** type, which is type **long** in **C**) of the same size of **A**.
- **C(k)=q** if **B(q,1)≤A(k)<B(q,2)**, and **C(k)=0** otherwise. That is, for each element in **A**, the corresponding element in **C** indicates the row index of the interval in **B** that contains its value.

The corresponding function in MATLAB is

```
function C = lab11m(A, B)
C = zeros(size(A), 'int32');
for q = 1:size(B,1)
  C(A >= B(q,1) & A < B(q,2)) = q;
end
```

You can use this code to check whether your MEX version works correctly. Note: You need to use different function names of the two versions. You can name your MEX function **lab11c**.

Be sure to do the necessary input argument checking.

When you have both versions working correctly, try to compare their execution time on the same inputs.

Part B: (only for non-CS majors)

Your task is to extend the MATLAB code above to partially color a gray-scale image. The function header becomes

```
function [C, D] = lab11m(A, B, M)
```

Input **A** is a 2-D gray-scale image. (You can take any RGB image and use **rgb2gray** to convert it to gray-scale for your experiments.)

Input **B** are the intervals as described in Part A.

Input **M** represents a color map (3 columns for RGB) with the same number of entries (rows) as **B**.

Output **C** has the same meaning as in part A.

Output **D** is a RGB image of the same size as **A** and **C**. Its values are defined as

```
D(y,x,:) = A(y,x)          if C(y,x) == 0
D(y,x,:) = M(C(y,x),:)     if C(y,x) != 0
```

You cannot use loops over **x** and **y** for computing **C** and **D**.