CS69011: Computing Lab-1
Assignment 1: Sorting of structures
August 07, 2023

==========Instructions==========

1. Part A contains 20 marks, and Part B contains 30 marks.
2. In the case of user input, assume only valid values will be passed as input.
3. Regarding submission: Create a separate C file for each part. -> *PartA.c, PartB.c*
Create a separate text file having the output from your C file given user input. -> *PartA_output.txt, PartB_output.txt*
Create a zip file of all these C files in the name <RollNo>_A1_PartAB.zip and submit it to Moodle.

==========

Consider the problem of organizing a conference with "n" presentations. Each presentation has a starting time, an ending time, a popularity score, and a presentation title. The goal is to maximize the total popularity score of the presentations an attendee can watch without any time conflicts.

Your input will be (i) the total number of presentations, (ii) starting time, ending time (separated by a space), popularity score, and presentation title of each presentation on a separate line. Note that the starting and ending time may not be in sorted order.

Implement the following strategies for solving this problem.

(a) A naive strategy is to sort the presentations by their starting times, and then iteratively select the presentation with the earliest starting time that does not conflict with any previously selected presentation without maximizing the total popularity score. Implement this algorithm. What is the complexity of this algorithm?

(b) A better strategy is to sort the presentations by their popularity scores in descending order and then iteratively select the presentation with the highest popularity score that does not conflict with any previously selected presentation. Implement this algorithm. What is the complexity of this algorithm?

Input format:

You can take input from a file with the following format:

<Number of presentations>

<Start time in HHMM format> <End time in HHMM format> <Popularity score> <String containing presentation title>
...
...

You can take the version of the algorithm to use (a/b/c), and the path of the file as command-line inputs.

Output format:
Print the list of selected presentations in one presentation per line, along with their starting times, ending times, and popularity scores.
