CSC/DCSCI 2720: Data Structures
Lab 6

Instructor: Shiraj Pokharel

Due : 48 hours after release
Late Submission deadline (with 25% penalty) 24 hours after due date

Answer the below questions. You must submit your responses as a SINGLE Jupyter Notebook, where each program is put in separate Jupyter Notebook cells within that SINGLE Jupyter Notebook. Do NOT submit Colab links. Failure to comply with this simple requirement will result in a score of Zero. Please, be careful not to be assigned a Zero score this way.

Few Rules to be followed, else will receive a score of ZERO

(1) Your submissions will work exactly as required.

(2) Your files shall not be incomplete or worse corrupted such that the file does not compile at all. Make sure you submit a file that compiles.

(3) Your submission will show an output. Should you recive a Zero for no output shown do not bother to email me with "but the logic is perfect" !

Note that your program's output must **exactly** match the specs(design , style) given here for each problem to pass the instructor's test cases.
Design refers to how well your code is written (i.e. is it clear, efficient, and elegant), while *Style* refers to the readability of your code (commented, correct indentation, good variable names).

In user content generated web-services - lets say YouTube - the process of de-duplication is of serious importance. One straight forward reason is same video by different names is just an extra cost in data-storage. So getting rid of exact duplicate content makes financial sense. Please be reminded that exact same content with different video qualities may not be candidates for removal via de-duplication.

Today we will explore on ways to do a de-duplication of videos where video filenames are presented as integers. For the purposes of our task, we will set a very narrow criterion for de-duplication : just the filenames.

Below is how the filenames are represented
ARRAY[] = [50, 11, 33, 21, 40, 50, 40, 21, 40]

Below is the expected output after de-duplication
ARRAY[] = [11, 21, 33, 40, 50]

ATTN : Notice the reduced size of the output array.
Also, take a look again at the input array and be reminded that the input array is not sorted !

Further, please be reminded that you cannot use library functions to either sort and or perform the de-duplication operation. Doing so would straight up result in a score of Zero !

You will solve the problem as:-

(1) [100 points] Implement the function in such a way that your solution could solve the problem with $O(n \, log_2(n))$ time complexity overall but stays within the bounds of $O(1)$ space complexity. Here, n is the length of the list of input integers (array). I believe the sorting routine that should be used here is **Quick Sort** with the last input element acting as the ”pivot”. Please state as code comment which sorting routine you are using, sort the array with that algorithm and solve the de-duplication problem thereafter. De-duplication part of the solution in itself must adhere to $O(n)$ time bound. However, at this stage of the course we will not be considering any memory used by recursion.

Very Very Important :

(1) Your code should be well commented which explains all the steps you are performing to solve the problem. A submission without code comments will immediately be deducted 15 points !

(2) As a comment in your code, please write your test-cases on how you would test your solution assumptions and hence your code.