In this project, you will implement various versions of the Quick Sort algorithm, execute these algorithms with different complexity cases and data sizes, report your results as execution times, and make comments on the results.

You will implement four versions of the Quick Sort algorithm as shown below:

Ver1. The classical deterministic algorithm. The pivot is chosen as the first element of the list.

Ver2. The randomized algorithm. The pivot is chosen randomly. This is the algorithm "Quicksort (1st version)" in the course slides.

Ver3. The randomized algorithm. The list is first randomly permuted and then the classical deterministic algorithm is called where the pivot is chosen as the first element of the list. This is the algorithm "Quicksort (2nd version)" in the course slides.

Ver4. The deterministic algorithm. The pivot is chosen according to the "median of three" rule.

In the executions, you will analyze two cases as shown below:

Case1. Average case. Inputs will be generated randomly.

Case2. Worst case. Input will be a sorted list.

The inputs for the algorithms will be in four different types as shown below. Suppose that the data size is n.

InpType1. Each element of the list is an integer between 1 and $10*n$. The elements are randomly chosen within this range. Note that the elements in the list will be mostly distant integers due to the interval used to choose the elements.

InpType2. Each element of the list is an integer between 1 and $0.75*n$. Note that in this case there will be duplicate elements in the list.

InpType3. Each element of the list is an integer between 1 and $0.25*n$. Note that in this case there will be much more duplicate elements in the list.

InpType4. All the elements are the integer 1.

You will consider three different data sizes as $n=100$, $n=1,000$, and $n=10,000$. While analyzing the average case, for each execution (e.g. Ver1, InpType1, $n=100$), you must execute the algorithm with random inputs several times and take the average in order to observe the average behavior – execute the algorithm 5 times. While analyzing the worst case, for each execution (e.g. Ver1, InpType1, $n=100$), you will execute the algorithm with only one input which is an already sorted list.

As a summary, you need to work on this project in the following order:

i)   First, implement the four versions of the algorithm.
ii)   Then, with data size $n=100$:
    a. For InpType1, generate five different inputs for average case analysis and one input for worst case analysis. Execute the four algorithms with the average inputs and the worst input.
       For each version of the algorithm, report the average case execution time (average of the five execution times) and the worst case execution time.
    b. Do the same (i.e. repeat item (a)) for InpType2, InpType3, and InpType4.
iii) Do the same (i.e. repeat item (ii)) with $n=1,000$ and $n=10,000$.

Submission:

- The program codes and a readme file that explains how to run the codes will be submitted.
- The answers must be submitted in the two answer sheets provided in Moodle as template (AnswerSheet1.docx and AnswerSheet2.docx). Delete the parts written in italics in these files and write the necessary results.
  Do not change the format in the answer sheets. You must write your results exactly in the specified format.
- In AnswerSheet1 document, all the inputs for all the data sizes and experiments must be written. The size of the file will be quite large due to the size of the input lists, but anyway all the inputs must be written.
- In AnswerSheet2 document, for each case (average case and worst case), the table and the comments must fill a whole page as shown in the template. The comments are an important part of the project and the project will not be accepted if detailed comments are not provided.
- Write the answers using a word processor; not in handwritten.
- Both students in the group MUST submit the project. The submissions of the two students (i.e. the code documents, the readme file, and the answer documents submitted by the two students) must be exactly the same. A submission by a student implies that she/he has contributed to nearly half of the submitted codes and answers.
- In addition to the project files, each student in a group must submit a document stating explicitly the parts of the project she/he has worked on. The document must begin with the line "I worked on the following parts in this project:" and all the parts the student has worked on must be explained very clearly and in not less than 10 lines. Do not write general comments such as "we worked on the project together", and the like. Write what you have done in the project explicitly. If both students write the same or similar explanations, their projects will not be graded.
- Upload a single zip file in Moodle, which consists of the program codes, readme file, answers, and the document that explains the parts of the project you worked on. Name the files as follows:
  - Program codes: NameSurname.xxx
  - Readme file: Readme.txt
  - Answer Sheet 1: NameSurnameAS1.pdf

- o   Answer Sheet 2: NameSurnameAS2.pdf
- o   Document: NameSurnameMyWork.pdf

Collect all these files into a single file NameSurname.zip that will be submitted.

- Each group must answer the exam themselves, without any interactions with others. The codes must be implemented by yourself. No materials from resources (internet, books, etc.) are allowed to be used.

Deadline:
- The deadline is 08.01.2023 Sunday 23:59. The deadline is strict.
- No late submission will be accepted.