# CMP3005 Analysis of Algorithms
# Term Project

Fall 2020

**Due date:** January 14, 2021 until 23:59

## Plagiarism Detector

# Introduction

In this project you will construct a system that checks the similarity rate between a specific document and a set of documents.

# Submission

- Submit all source code through **itslearning**.
- In addition to the source code, you should submit a min 3 pages report in a separate file which includes the methods (string matching, hashing, searching, etc) you have applied, the programming language (only **C++** or **Java** is allowed) you have used, the libraries you have used, the average speed of your algorithm in terms of milliseconds, exemplary outputs for given documents, worst case Big-OH complexity of your algorithm, etc.
- The system will automatically be closed at the specified deadline: January 14, 2021 at 23:59 and the submissions after that time **will NOT be accepted**.

# 1 Implementation

## 1.1 Input files

The program will get two inputs:

1) A set of documents (.txt files) under a folder,
2) The main file (.txt file) which will be checked for plagiarism.

## 1.2 Outputs

The program will display the following outputs:

1) The similarity rate between the main document and each document under the folder given as input,

2) The most similar 5 statements / sentences for each document.

For the evaluation of your implementation, during the demo that will be held in the last week of the semester you will be asked to run your program for a folder containing multiple .txt files named as 'document1.txt', document2.txt', etc. and a main .txt file called "main_doc.txt".

## 1.3 Details

- You can use any text searching algorithm and/or data structure you would like; you can even use algorithms not discussed in class.
- The pattern matching algorithm must be written by yourself.
- In addition to your algorithm's similarity detection ability, you will also be graded on the speed of your code, so you should try to choose an efficient algorithm.

## 1.4 Important Instructions

Your program should give an output to the console in the following format when executed:

*1)* {Similarity Rate with the First Document}
{Most Similar Sentence}
{Second Most Similar Sentence}
{Third Most Similar Sentence}
{Fourth Most Similar Sentence}
{Fifth Most Similar Sentence}

*2)* {Similarity Rate with the Second Document}
{Most Similar Sentence}
{Second Most Similar Sentence}
{Third Most Similar Sentence}
{Fourth Most Similar Sentence}
{Fifth Most Similar Sentence}

n) {Similarity Rate with the $n^{th}$ Document}
{Most Similar Sentence}
{Second Most Similar Sentence}
{Third Most Similar Sentence}

{Fourth Most Similar Sentence}
{Fifth Most Similar Sentence}

## 1.5 Grading

You will be graded on the run time of your algorithm, so it is expected that your algorithm runs efficiently. Also, the accuracy of your program (to what extent similarities are found) will be taken into account.

30% Running time, efficiency

25% Detection Ability (to what extent similarities can be found)

25% Report

20% Code structure (readability and reusability of the implementation, the programming style used, comments, etc)

## 1.6 Submission Instructions

- You must work in groups of 3 people. Individual projects and gropus with less than or more than 3 students will NOT be accepted.
- You will present your project at the last week of the semester and explain all details of the methods you used. If you cannot explain your algorithm and answer the questions in detail during the demo, you will get 0 (zero) from the project and disciplinary actions will be taken for all the students in the project group.
- Implementation can be done with C++ or Java.
- Do NOT use built-in text search functions of the programming language you used, implement it yourself.
- Submit your source code and report through itslearning as a .zip file.
- Only one person out of each group should submit the project. The file name should include the student id of the student that has submitted the project in the following format:
  {STUDENT NUMBER}.rar

## Cheating Policy

- Cheating is strictly prohibited.
- If you cannot explain your algorithm and answer the questions in detail during the demo, you will get 0 (zero) from the project and disciplinary actions will be taken for all the students in the project group.

- Everything must be your own work, do not use each other's source code. If cheating is confirmed all students involved **will be penalized heavily.**

**Important:** itslearning has built-in plagiarism control that automatically detects submitted material that is plagiarised.