

## 1 Introduction

In this project, you are given a programming question, and you will develop an efficient algorithm and implement it. You can use any of the programming languages you prefer, there is no limitation. However, usage of any libraries except the standard language library are forbidden. Please request permission if you plan to use any of these (if you are using Python, NumPy is allowed).

Please open the [Google Sheet Link](#), and enter your group number, names of the members and the emails of the members to the cell, where your project is listed. Please keep in mind that the sheet document is divided into different sub-sheets. Thus, select the correct sub-sheet with respect to your group number. All of the test cases for each of the projects can be viewed and downloaded from this [Google Drive Link](#).

Please open a GitHub public repository, include all of your members as contributors and add the repository link to the given Google Sheet document. This step is quite important for us to see your progress and has to be done quickly. Therefore, in the README file please keep a list of completed steps, a TO DO list and the results retrieved if there are any. You will also prepare a presentation and present to the TAs. Therefore, you should also create a Google Slides presentation and include its link to the Google Sheet document.

Please email to [comp305staff-group@ku.edu.tr](mailto:comp305staff-group@ku.edu.tr), if there is any problem in viewing the drive folder or modifying the document or if you have some troubles with any of the test cases.

## 2 Presentation Details

Each of the presentations should take  $\sim 10$  minutes and there will be a 5-minute Q&A session afterwards. If a presentation lasts longer than 10 minutes, then it will be interrupted. During the presentation each of the groups should explain and report:

- The algorithm you designed to solve the problem, the choices of the data structures you used and your reasoning.
- The time complexity of your algorithm (and the space complexity if applicable).
- Your run times for each of the test cases.
- Further improvements that can be done as future works.

This project does not expect from you to come up with just one solution and then test only that solution. For each of the problems you can start with some baseline approaches with more complexity and improve the baseline algorithm step by step. Be as creative as possible. Report different approaches you tested and why did you decide on the final algorithm you present. Your grading will be based on your creativity, your cumulative progress and how well did you present your approach. Additionally, there will be more test cases which won't be provided to you. Therefore, to be sure about the correctness of your algorithm, you should create extra test cases. Finally, having an efficient and fast algorithm is also as important as having correct algorithm. Do your best to find the most efficient and fastest solution. Your effort is really important for grading.

### 3 Deadlines

You can work on your project until the end of *23th May, 2021*. The project presentations will be held between *24th-28th of May, 2021*.

In the following pages, you can see the available project(s):

# Discovery of Atiye

The rock formations that have created Cappadocia is still one of the most popular monuments in Turkey. It was created with a geological process that began millions of years ago and today, it is still a popular destination for many visitor. It is known to have underground cities and tunnels and many of them are accessible to tourists. However there are rumors that it contains some secret and newly discovered places. The rumors started when the crew of a new Turkish series, Atiye, had accidentally discovered a new cave during the shooting of a scene. Even though, the cave was seen like every other at first sight, after certain inspections by experts, some hidden and ancient texts were found, carved in the walls. Nonetheless, the text remained meaningless for years since it was written with a secret code. Last year, in Göbeklitepe, the researchers discovered a key to translate the texts. The key was a document with descriptions on how to convert the letters. The challenging part is that, there can be more than one translation or no translation for a letter. If the researchers succeed in converting the letters according to the key, they might gain some insight on the communities that lived there but they ask for your help. They are sharing with you some parts of the key to how to convert the letter and a small list of word pairs containing the original and new founded words. You are asked to check if the two words are equal to each other. In order to call two words equal there are two criteria: 1) Their length should be equal 2) You should be able to convert all the letters in the first word to the parallel letter in the second word with the given translations. You can use the translations zero or more time.

## 4.1 Input and Output

As input, first of all you have two numbers. First number in the line,  $x$ , shows the number of translations of letters you are given. Second number,  $y$ , is the number of word pairs you will work on. For  $x$  lines you will see a pair of letters with space between them. The letters represent the possible translations. For instance, if the letter pair is (c m), you should understand that you can convert the letter c to m. You will see each pair only once. After  $x$  lines, meaning once the letter pairs are finished, for  $y$  lines there will word pairs. You are expected to check if two words match each other. You can only use lowercase letters. Words have at least 1 letter and no more than 50 letters.

## 4.2 Sample Input Output

Sample Input 1	Sample Output 1
9 5 c t i r k p o c r o t e t f u h w p we we can the work people it of out the	yes no no yes yes
Sample Input 2	Sample Output 2
3 3 a c b a a b aaa abc abc aaa acm bcm	yes no yes

Figure 1: Sample Input and Output File.