

CHALLENGE 01: STRING MATCHING

1 Introduction

1.1 Research

String matching is a branch of searching algorithms, which is widely applied in many different scientific fields. See **chapter 32, Introduction to Algorithms** (*3rd edition*) and fulfill the following requirements:

1. Define the *String matching* problem, consists of inputs (and their conditions), and problem requirements. Outline 2 applications of *String matching* in real life, point out **inputs** and **outputs**, respectively.
2. Explain the algorithms, step-by-step example, point out the time-complexity (with explanation) and make a comparison table for the following *String matching* algorithms:
 - Brute-force (Naive String-matching)
 - Rabin-Karp
 - Knuth-Morris-Pratt

1.2 Programming

- **Crossword game:** Given a characters table with size $W \times H$ see **Figure 1**. Write an illustration program getting the strings of meaningful words from a file. A string is considered FOUND if its represented word appears in a row from left to right or in a column, from up to down. Determine whether the strings are FOUND and indicate the position and direction of the found words.
- **Input:** Input file *input.txt*:
 - 1st line: 2 integers W and H , separated at least by a space " ", represent Width and Height of the table of characters, respectively.
 - Next H lines: each consists of W characters, separated from each other by a space " ".
 - From the $(H + 2)^{th}$ lines: Each line contains a string (representing a meaningful word), which needs to be matched.
 - Final line is represented by #.

- 2019/CS1.png

	1	2	3	4	5	6	7	8	9	10
1	U	C	M	A	R	V	E	L	O	L
2	S	F	D	Q	U	E	U	E	G	I
3	T	R	A	V	E	R	S	A	L	N
4	E	F	G	S	T	A	X	F	N	K
5	K	D	P	U	Z	U	V	U	C	E
6	I	S	I	B	L	I	N	G	S	D
7	E	N	N	T	S	L	Y	C	L	L
8	D	E	G	R	E	E	A	B	I	I
9	G	X	Z	E	M	O	Q	R	F	S
10	E	R	T	E	B	G	R	O	O	T

Figure 1: Table of characters

• Output:

- 1st line: Number of words found.
- Next lines: Follow the structure [**Word**_**Pos**_**Dir**] (_ representing a space (" ")). In which:
 - * **Word** is the given word.
 - * **Pos** is the coordinate of the given word, represent by (x, y) (x, y ≥ 1). Return (0, 0) for non-existing words.
 - * **Dir** is the direction of the given word: **LR** if the direction is from left to right, **TD** if the direction is from up to down. Return **NF** for not-found words.

• Example:

Input	Output
9 10 U C ... L ... E R ... T MARVEL LIST XXX #	2 MARVEL (1,3) LR LIST (7,10) TD XXX (0, 0) NF

2 Group registration and Submission regulations

2.1 Group Registration

- This challenge requires a group of 3 - 4 students. Group members for each challenge must be different (i.e, Any pairs of students are at max ONE same group).
- Group ID is generated by concatenating the last 2 digits of each member's Student ID in ascending order.

Example:

– Given the student codes: *19127666 - 19127888 - 19127991 - 19127999*.

→ **Generated ID:** *66889199*.

– Given the student codes: *19127667 - 19127889 - 19127990*

→ **Generated ID:** *678990*.

- Group registration will be provided within the attached [link](#). **Each group's member** must fill in the registration form.

Note: Group registration and file submission time should not be different more than *15mins*.

2.2 Submission regulations

- Only 12 first submissions is accepted. From the 13th is illegal.
- The submission file must be in the following format: [**Group_ID.rar**] or [**Group_ID.zip**], is the compression of the [**Group_ID**] folder. This folder contains:
 - The report file must be presented as a document [**Group_ID.pdf**] or as a slideshow [**Group_ID.pptx**]. This file presented research answers from **1.1** and the information of code fragment (data structures, algorithms, functions) from **1.2**.
 - * If your submission is a slideshow, there must be explanation in the *Note* part of each slide.

- * Information (Names, Student IDs) must be declared clearly on the first page (or first slide) of your report. Your working progress (Which option did you choose? How much work have you completed?) should be demonstrated on this page, too.
- * The report file should be **structured, logical, clear** and **coherent**. The length of the submission should not exceed 15 pages for the document file, and 30 pages for the presentation slide.
- * All links and books related to your submission must be mentioned.
- The programming file must be a single file [**Group_ID.cpp**]. The code fragment must be clear, logical and commented.
- Submission with wrong regulation will result in a "0" (zero).
- Plagiarism and Cheating will result in a "0" (zero) for the entire course and will be subject to appropriate referral to the Management Board of the CLC program for further action.