

Project: TRIES

1 Requirements

1.1 Theory Part

Trie is an efficient *information re***Trie**val data structure. Using *Trie*, searching operation can be brought to optimal limit (key length). Using the code given (**Tries.pdf**) as the reference, students are requested to fulfill the following requirements:

1. Show the time complexity of the following operations (of *Trie*):
 - Adding a word.
 - Removing a word.
 - Searching a word.
 - Searching words which has the same prefix with length i .
2. Show the advantages of *Trie* comparing to other data structures (designed for searching) which you have learned: **Binary search tree** and **Hash table**.

1.2 Programming Part

1. Students implement and build a Trie (with words from the given files, including):
 - File *Dic.txt* contains a list of English words sorted ascending. Each word locates on a single line. You have to use this list of words to build the Trie.
 - File *Tries.pdf* contains the source code for operations on the Trie data structure. **You do not need to use source code from this file.**

2. With the built *trie* (containing the provided English words), implement a program to generate a list of *valid English words*¹ which have letters from a given character list. (*Note: Valid English words do not require to have all of provided letters but must have at least 3 letters*).

- **Input:** List of letters use for creating valid English words. These letters must be on the same line and satisfied the following requirements:
 - Being in lowercase, not in CAPITAL.
 - Being sorted ascending in lexicographic order, separated by a single space " ".
 - May appear multiple times.
- **Output:**
 - The 1st line: An integer N indicates the number of created words.
 - Next N lines: each line contains a created word.
- **Example:**

Input	Output
a c e p	6 ace ape cap cape pace pea

¹valid English words are words existed in the given dictionary file.

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 10 first submissions is accepted.
- 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 referenced.
- 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).
- Submission without registration will result in a "0" (zero) for ALL group members.
- 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.

END