# Company User Feedback - Search Algorithm

## Question

A company is building a way to help users search reviews quicker by providing real-time suggestions to search terms when the user starts typing. When given a minimum of two characters into the search field the system will suggest at most three keywords from the review word repository. As the user continues to type in the reviews search bar the relevant keyword suggestions will update automatically.

Write an algorithm that will output a maximum of three keyword suggestions after each character is typed by the user in the search field.

If there are more than three acceptable keywords, return the keywords that are first in alphabetical order. Only return keyword suggestions after the user has entered two characters. Keyword suggestions must start with the characters already typed. Both the reviewsRepository and the userInput should be compared in a case-insensitive way.

* **Input** - The input to the method/function consists of two arguments: reviewsRepository (a list of unique strings representing the various keywords from the company review comment section) and userInput (string representing the full search query of the user).
* **Output** - Return a list of a list of strings in lower case, where each list represents the keyword suggestions made by the system as the user types each character of the userInput. Assume the user types characters in order without deleting or removing any characters. If an output is not possible, return an empty array.

Please provide written explanation of your approach.

## Bonus task

Provide time and space complexity of your approach.

*\*Time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the input. Similarly, space complexity of an algorithm quantifies the amount of space or memory taken by an algorithm to run as a function of the length of the input. Time and space complexity depends on lots of things like hardware, operating system, processors, etc. However, we don't consider any of these factors while analyzing the algorithm. We will only consider the execution time of an algorithm.*

## Example

Input: reviewsRepository = [ "mobile", "mouse", "moneypot", "monitor", "mousepad" ] userInput = "mouse"

Output: ["mobile", "moneypot", "monitor"] ["mouse", "mousepad"] ["mouse", "mousepad"] ["mouse", "mousepad"]

Explanation: The chain of words that will generate in the search box will be mo, mou, mous, mouse and each line from output shows the suggestion of "mo", "mou", "mous", "mouse", respectively in each line. For the keyword suggestions made by the system that are generated for 'mo', the matches that will be generated are: ["mobile", "mouse", "moneypot", "monitor", "mousepad"] Alphabetically, they will be reordered to [ "mobile", "moneypot", "monitor", "mouse", "mousepad" ]. Thus the keyword suggestions made by the system are ["mobile", "moneypot", "monitor"].

## Preferred technologies

We prefer that you use C# or Java for the implementation. The below starting code is in C# which can be easily adjusted to Java.

## Code

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

public class Foo

{

/\*

\* Complete the 'SearchSuggestions' function below.

\*

\* The function is expected to return a list of string lists (2D\_STRING\_ARRAY).

\* The function accepts following parameters:

\* 1. List (STRING\_ARRAY) - reviewsRepository

\* 2. String - userInput

\*/

public static List<List<string>> SearchSuggestions(List<string> reviewsRepository, string userInput)

{

}

}

public class Solution

{

public static void Main(string[] args)

{

int reviewsRepositoryCount = Convert.ToInt32(Console.ReadLine().Trim());

List<string> reviewsRepository = new List<string>();

for (int i = 0; i < reviewsRepositoryCount; i++)

{

string reviewsRepositoryItem = Console.ReadLine();

reviewsRepository.Add(reviewsRepositoryItem);

}

string userInput = Console.ReadLine();

List<List<string>> foo = Foo.SearchSuggestions(reviewsRepository, userInput);

Console.WriteLine(String.Join("\n", foo.Select(x => String.Join(" ", x))));

}

}