▾ Lab#2, NLP@CGU Spring 2023

This is due on 2023/03/13 15:30, commit to your github as a PDF (lab2.pdf) (File>Print>Save as PDF).

IMPORTANT: After copying this notebook to your Google Drive, please paste a link to it below. To get a publicly-accessible link, hit the *Share* button at the top right, then click "Get shareable link" and copy over the result. If you fail to do this, you will receive no credit for this lab!

LINK: paste your link here

https://colab.research.google.com/drive/xxxxxxxxx
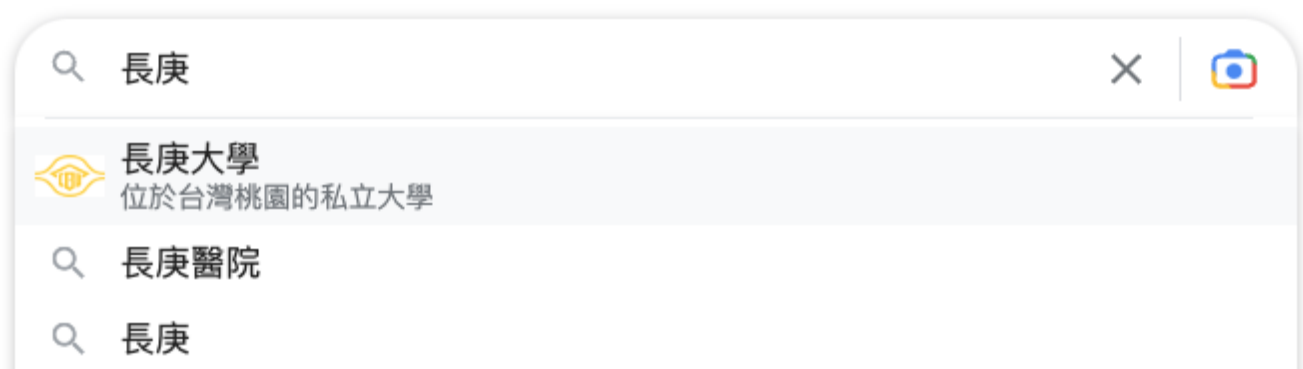

此內容會顯示為程式碼

Student ID: B0928006

Name: 廖采葳

▾ Question 1 (100 points)

Implementing Trie in Python.

Trie is a very useful data structure. It is commonly used to represent a dictionary for looking up words in a vocabulary.

For example, consider the task of implementing a search bar with auto-completion or query suggestion. When the user enters a query, the search bar will automatically suggests common queries starting with the characters input by the user.

位於台灣桃園的科技大學

🔍　長庚掛號查詢

按兩下 (或按 Enter 鍵) 即可編輯

```python
# YOUR CODE HERE!
# IMPLEMENTIG TRIE IN PYTHON
class TrieNode:
    def __init__(self, char):
        self.char = char
        self.children = {}
        self.finished = False
        self.count = 0

class Trie:
    def __init__(self):
        self.root = TrieNode("")

    def insert(self, word):
        node = self.root
        for char in word:
            if char in node.children:
                node = node.children[char]
                node.count += 1
            else:
                new_node = TrieNode(char)
                node.children[char] = new_node
                node = new_node
                node.count += 1
        node.finished = True

    def dfs(self, node, prefix, results):
        if node.finished:
            results.append((prefix, node.count))
        for child in node.children.values():
            self.dfs(child, prefix+child.char, results)

    def query(self, prefix):
        node = self.root
        for char in prefix:
            if char in node.children:
                node = node.children[char]
            else:
                return []
        results = []
        self.dfs(node, prefix, results)
```

```
        return sorted(results, key=lambda x: -x[1])

# create Trie object
```

```
# # DO NOT MODIFY THE VARIABLEStrie.insert("長庚資工")
obj = Trie()
obj.insert("長大")
obj.insert("長庚")
```

8 1 of 6 problems

"trobjie" is not defined (reportUndefinedVariable)

```
obj.insert("長庚")
obj.insert("長庚大學")
obj.insert("長庚科技大學")
```

```
# # DO NOT MODIFY THE BELOW LINE!
# # THE RESULTS : [(words, count), (words, count)]
print(obj.query("長"))
# [('長庚', 2), ('長庚資工', 1), ('長庚大學', 1), ('長庚科技大學', 1), ('長大', 1)]

print(obj.query("長庚"))
# [('長庚', 2), ('長庚資工', 1), ('長庚大學', 1), ('長庚科技大學', 1)]

        [('長庚', 5), ('長庚資工', 1), ('長庚大學', 1), ('長庚科技大學', 1), ('長大', 1)]
        [('長庚', 5), ('長庚資工', 1), ('長庚大學', 1), ('長庚科技大學', 1)]
```

Colab 付費產品 － 按這裡取消合約

Colab 付費產品 － 按這裡取消合約