

trie & hash

Trie：搜尋系統的前綴熱門字查詢 (Autocomplete with Frequencies)

```
class TrieNode:
    def __init__(self):
        self.children = {}    # dict: char → TrieNode
        self.is_end = False  # 是否是一個完整單字的結尾
class Trie:
    def __init__(self):
        self.root = TrieNode()
```

1：ADD word

將單字 word 加入系統。

如果 word 已存在，代表它被搜尋一次，頻率 +1。

2：QUERY prefix k

查詢所有以 prefix 開頭的單字，並將它們依**搜尋頻率降序**排序。

若頻率相同，使用**字典序**排序。

只回傳前 **k** 個字。

3：REMOVE word

刪除字典中的 word（如果存在），減少它的頻率：

- 若目前頻率 > 1：頻率 -1
 - 若頻率 == 1：完全移除
 - 若不存在：忽略
-

Hash : Two Sum (兩數和)

給你一個整數陣列 `nums` 和一個目標值 `target`。

請找出陣列中 **兩個不同的元素** 使得：

$$\text{nums}[i] + \text{nums}[j] = \text{target}$$

並回傳其中 **任意一組** `(i, j)` 的索引。

範例輸入

```
nums = [2, 7, 11, 15]  
target = 9
```

範例輸出

```
(0, 1)
```