

CS2023 - Aula de Ejercicios N° 9
Brenner H. Ojeda Rios
Semestre 2024-0

Se sugiere que cada estudiante trate de resolver los ejercicios de forma **individual** y luego los discuta en grupo. **En cada ejercicio debe usar la estructura de datos Trie.**

Ejercicios

1. (5 pts) Escriba una función para encontrar la cadena de prefijo común más larga entre un arreglo de strings.

Si no hay un prefijo común, devuelve una cadena vacía.

- **Ejemplo 1:**

Input: `strs = ['flower', 'flow', 'flight']`

Output: `'fl'`

- **Ejemplo 2:**

Input: `strs = ['dog', 'racecar', 'car']`

Output: `''`

Explicación: No existe un prefijo común entre las cadenas de entrada.

Restricciones:

- $1 \leq \text{strs.length} \leq 200$
- $0 \leq \text{strs}[i].\text{length} \leq 200$
- `strs[i]` consta únicamente de letras minúsculas en inglés.

2. (6 pts) Dada una arreglo de cadenas de palabras y un número entero k , devuelve las k cadenas más frecuentes.

Devuelve la respuesta ordenada por frecuencia de mayor a menor. Ordena las palabras con la misma frecuencia por su orden lexicográfico.

■ **Ejemplo 1:**

Input: `words = ['i', 'love', 'lourdes', 'i', 'love', 'coding'], k = 2`

Output: `['i', 'love']`

Explicación: 'i' y 'love' son las dos palabras más frecuentes. Tenga en cuenta que 'i' viene antes de 'love' debido a un orden alfabético inferior.

■ **Ejemplo 2:**

Input: `words = ['the', 'day', 'is', 'sunny', 'the', 'the', 'the', 'sunny', 'is', 'is'], k = 4`

Output: `['the', 'is', 'sunny', 'day']`

Explicación: 'the', 'is', 'sunny' y 'day' son las cuatro palabras más frecuentes, siendo el número de apariciones 4, 3, 2 y 1 respectivamente.

Restricciones:

- $1 \leq \text{words.length} \leq 500$
- $1 \leq \text{words}[i].\text{length} \leq 10$
- `words[i]` consta de letras minúsculas en inglés.
- k está en el rango `[1, el número de words[i] únicas]`.

3. (9 pts) Se le proporciona una arreglo de palabras de tamaño n que consta de cadenas no vacías.

Definimos la **puntuación** de un string `word` como el **número** de strings `words[i]` tales que `word` es un prefijo de `words[i]`.

- Por ejemplo, si `words = ['a', 'ab', 'abc', 'cab']`, entonces la puntuación de `'ab'` es 2, ya que `'ab'` es un prefijo tanto de `'ab'` como de `'abc'`.

Devuelve un arreglo `answer` de tamaño n donde la `answer[i]` es la suma de las puntuaciones de cada prefijo no vacío de `words[i]`.

Tenga en cuenta que una cadena se considera un prefijo de sí misma.

■ **Ejemplo 1:**

Input: `words = ['abc', 'ab', 'bc', 'b']`

Output: `[5,4,3,2]`

Explicación: La respuesta para cada cadena es la siguiente

- `'abc'` tiene 3 prefijos: `'a'`, `'ab'`, and `'abc'`.
- Hay 2 cadenas con el prefijo `'a'`, 2 cadenas con el prefijo `'ab'`, y 1 cadena con el prefijo `'abc'`. El total es `answer[0] = 2 + 2 + 1 = 5`.
- `'ab'` tiene 2 prefijos: `'a'` y `'ab'`.
- Hay 2 cadenas con el prefijo `'a'`, y 2 cadenas con el prefijo `'ab'`. El total es `answer[1] = 2 + 2 = 4`.
- `'bc'` tiene 2 prefijos: `'b'` y `'bc'`.
- Hay 2 cadenas con el prefijo `'b'`, y 1 cadena con el prefijo `'bc'`. El total es `answer[2] = 2 + 1 = 3`.
- `'b'` tiene 1 prefijo: `'b'`.
- Hay 2 cadenas con el prefijo `'b'`. El total es `answer[3] = 2`.

■ **Ejemplo 2:**

Input: `words = ['abcd']`

Output: `[4]`

Explicación: `'abcd'` tiene 4 prefijos: `'a'`, `'ab'`, `'abc'`, y `'abcd'`. Cada prefijo tiene un puntaje de uno, así la respuesta total es `answer[0] = 1 + 1 + 1 + 1 = 4`.

Restricciones:

- $1 \leq \text{words.length} \leq 1000$
- $1 \leq \text{words}[i].\text{length} \leq 1000$
- `words[i]` consta de letras minúsculas en inglés.