**Topic 6.17: Universal Words from String Arrays**

**Question**  
You are given two string arrays words1 and words2.  
A string b is a subset of string a if every letter in b occurs in a including multiplicity.

Example:

* "wrr" is a subset of "warrior" but not of "world".

A string a from words1 is **universal** if for every string b in words2, b is a subset of a.

Return an array of all the universal strings in words1. You may return the answer in any order.

**Aim**  
To identify universal strings from one list that contain all the character requirements defined by another list of strings.

**Algorithm**

1. For each word in words2, compute the maximum count of each character across all words in words2.
   * Example: For ["e", "o"], we require at least one e and one o in any universal word.
2. For each word in words1, count its characters.
3. Check if the word satisfies the maximum frequency requirements of all characters from step 1.
4. If it does, add it to the result list.
5. Return the final list of universal words.

**Output**A screenshot of a computer

AI-generated content may be incorrect.**S**

**Result**  
The algorithm successfully identifies universal strings in words1 that satisfy all subset conditions defined by words2.

**Performance Analysis**

* Time Complexity: O(m × k + n × k), where m = length of words2, n = length of words1, and k = average word length.
* Space Complexity: O(26) = O(1), for storing character frequency requirements.