# Problem 692: Top K Frequent Words

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 59.75%
**Paid Only:** No
**Tags:** Array, Hash Table, String, Trie, Sorting, Heap (Priority Queue), Bucket Sort, Counting

## Problem Description

Given an array of strings `words` and an integer `k`, return _the_`k` _most frequent strings_.

Return the answer **sorted** by **the frequency** from highest to lowest. Sort the words with the same frequency by their **lexicographical order**.

**Example 1:**

**Input:** words = ["i","love","leetcode","i","love","coding"], k = 2 **Output:** ["i","love"] **Explanation:** "i" and "love" are the two most frequent words. Note that "i" comes before "love" due to a lower alphabetical order.

**Example 2:**

**Input:** words = ["the","day","is","sunny","the","the","the","sunny","is","is"], k = 4 **Output:** ["the","is","sunny","day"] **Explanation:** "the", "is", "sunny" and "day" are the four most frequent words, with the number of occurrence being 4, 3, 2 and 1 respectively.

**Constraints:**

* `1 <= words.length <= 500` * `1 <= words[i].length <= 10` * `words[i]` consists of lowercase English letters. * `k` is in the range `[1, The number of **unique** words[i]]`

**Follow-up:** Could you solve it in `O(n log(k))` time and `O(n)` extra space?

## Code Snippets

### C++:

```
class Solution {
public:
vector<string> topKFrequent(vector<string>& words, int k) {

}
};
```

### Java:

```
class Solution {
public List<String> topKFrequent(String[] words, int k) {

}
}
```

### Python3:

```
class Solution:
def topKFrequent(self, words: List[str], k: int) -> List[str]:
```