Anagram Checking Algorithm

Problem Statement

We need to check whether two given words are anagrams, meaning they contain exactly the same characters but possibly in a different order. For example, the words 'tea' and 'eat' are anagrams because they consist of the same letters arranged differently.

Design the Algorithm

Algorithm:

1. Input two strings word1 and word2.
2. If their lengths are not equal, return False.
3. Sort both strings alphabetically.
4. Compare the sorted strings.
5. If they are identical, return True; otherwise, return False.

Pseudocode

```
Bool isAnagram(string word1, string word2) {
   if (word1.size() != word2.size()) {
      return false;
   }
   if (sorted(word1) == sorted(word2)) {
      return true;
   }
   else {
      return false;
   }
}
```

Analyze the Complexity

Operation	Complexity
Sorting each string	O (n log n)
Comparison	O(n)
Total Time Complexity	O(n log n)
Space Complexity	O(n)

Conclusion:

This algorithm correctly determines whether two words are anagrams using sorting and comparison, providing an efficient solution with a time complexity of O (n log n).