Test Reasoning 1

You are given a function named backwards that performs two operations on a vector of strings:

1. Reverses each individual string in the vector.
2. Sorts the reversed strings in descending lexicographical order.

The following test cases are provided:

* **Test Case 1**:  
  Input: ["apple", "banana", "cherry"]  
  Expected Output: ["elppa", "ananab", "yrrehc"]  
  (Explanation: Each string is reversed, and the reversed strings are sorted in descending order.)
* **Test Case 2**:  
  Input: [] (an empty vector)  
  Expected Output: []  
  (Explanation: An empty input vector remains unchanged after processing.)
* **Test Case 3**:  
  Input: ["hello"]  
  Expected Output: ["olleh"]  
  (Explanation: A single string is reversed but no sorting is needed as there is only one element.)
* **Test Case 4**:  
  Input: ["aaa", "aaa", "aaa"]  
  Expected Output: ["aaa", "aaa", "aaa"]  
  (Explanation: Strings remain the same after reversing, and sorting has no effect as all elements are identical.)

Explain why the backwards function produces these outputs for each of the given test cases. How does the function handle edge cases such as an empty vector or a vector with identical strings? What is the time complexity of the function, considering both the reversal and sorting operations? Why is this function suitable for the described tasks?