Gaming fan's nadir

Two strings are anagrams of one another if, ignoring capitalisation, punctuation and whitespace they contain the same characters. For instance `Finding Anagrams` and `Gaming fan's nadir` are anagrams. Utilities such as I rearrangement servant can help find anagrams which arise most commonly in cryptic crossword puzzles but also in other sorts of wordplay. The intent of this étude is to ask you to reproduce some of the functionality of such utilities.

Specifically, given a dictionary and a string you're going to be asked to find the "best" anagram for the string from words in the dictionary. The definition of "better" (and hence "best") in this context is as follows.

- Anagrams using fewer words are better than ones using more words.
- If two anagrams use the same number of words, they should first be compared by word length — if the longest words differ in length, then the one with a longer first word is better. If the longest words are equal in length but the second longest words differ in length then the one with a longer second longest word is better. And so on.
- If two anagrams have the same number of words, and corresponding words have the same length, then the earlier one in alphabetical order is better.

Task

Write a program that finds the best anagrams for each of a number of strings from a given dictionary. For the purpose of the following description, a "word" is just a string whose characters come from the range `a-z` (i.e., lower case letters). Input from `stdin` will be of the following form:

- a series of lines consisting of words,
- an empty line (or one consisting only of whitespace),
- another series of words.

The first group of words in the input are the words for which we seek anagrams. The second group (after the empty line) are the source dictionary which might consist of up to 100,000 words.

Output should be to `stdout`. For each word from the first group (and in the same order as they were input) a single line consisting of that word, a colon, a space, and then the best anagram for that word from the dictionary. If there are no anagrams for the word from the dictionary then the word, colon and space should still be printed.

For instance if the input is:

```
apple
appleapple
frog

app
el
```

Then the output should be:

```
apple: app el
appleapple: app app el el
frog:
```

Relates to Objectives

1.2, 1.4, 2.2, 2.3, 2.7, 2.9, 3.3, 3.4, 3.5, 3.6, 4.3.

(Pair)