In genetic research, sequencing a genome involves finding the exact sequence of base pairs (ATCG) present in the DNA of an organism. The DNA is often broken into short strands, each sequenced to obtain the base pair sequences. The whole genome is reconstructed using these partial sequences by exploiting the overlaps between strands.

Melbo is fascinated with genetics due to its applications like synthetic biology and CRISPR. To gain experience, Melbo wants to solve an NLP version of the sequencing problem. For a string, a character bigram is a sequence of 2 adjacent characters. For example, the word "area" has the bigrams ('ar', 're', 'ea').

Melbo wants to guess the word given only the bigrams. However, the bigrams are not presented in order. Instead:

1. The set of bigrams is calculated for a word.
2. Duplicates are removed.
3. The bigrams are sorted in lexicographic order.
4. Only the first 5 bigrams are retained.

For example, for the word "optional," the bigrams are ('op', 'pt', 'ti', 'io', 'on', 'na', 'al'). Sorted and trimmed, the set is ('al', 'io', 'na', 'on', 'op').

It may not be possible to uniquely identify a word given its bigrams due to:

1. Sorting introduces clashes: e.g., 'ar', 'ea', 're' for "area" and "rear".
2. Neglecting duplicates introduces clashes: e.g., 'be', 'de', 'es', 'id', 'si' for "beside" and "besides".
3. Retaining only 5 bigrams introduces clashes: e.g., 'al', 'io', 'na', 'on', 'op' for "optional" and "proportional".

Each bigram set corresponds to at most 5 words. For example, 'ci', 'en', 'ff', 'fi', 'ic' for "insufficient", "sufficient", and "sufficiently".

Melbo's task: Given a list of bigrams as a Python tuple with 5 or fewer sorted bigrams, guess one or more corresponding words. Melbo can make up to 5 guesses. If the correct word is in the guess list, Melbo scores 1

divided by the number of guesses made. If not, the score is 0. The average precision across all test points is the final precision of the model.