

Let n denote the number of strings, ℓ denote the maximum length of a string, and m denote the total length of the dictionary.

This is the word break problem with multiple queries, so it is at least as hard as 139. Word Break (the all-targets version).

1. Using the algorithm for word break, we can get $\tilde{O}(m^{4/3})$ running time.
2. DP+trie. $O(m\ell)$.
3. Aho-Corasick automaton. $O(m \cdot \frac{\ell}{w})$ (using bit packing), or $O(m\sqrt{m})$ (or $O(m \cdot \frac{\sqrt{m}}{\sqrt{w}})$ using bit packing on matching words with length $O(\sqrt{m} \cdot \sqrt{w})$).

44 / 44 test cases passed.

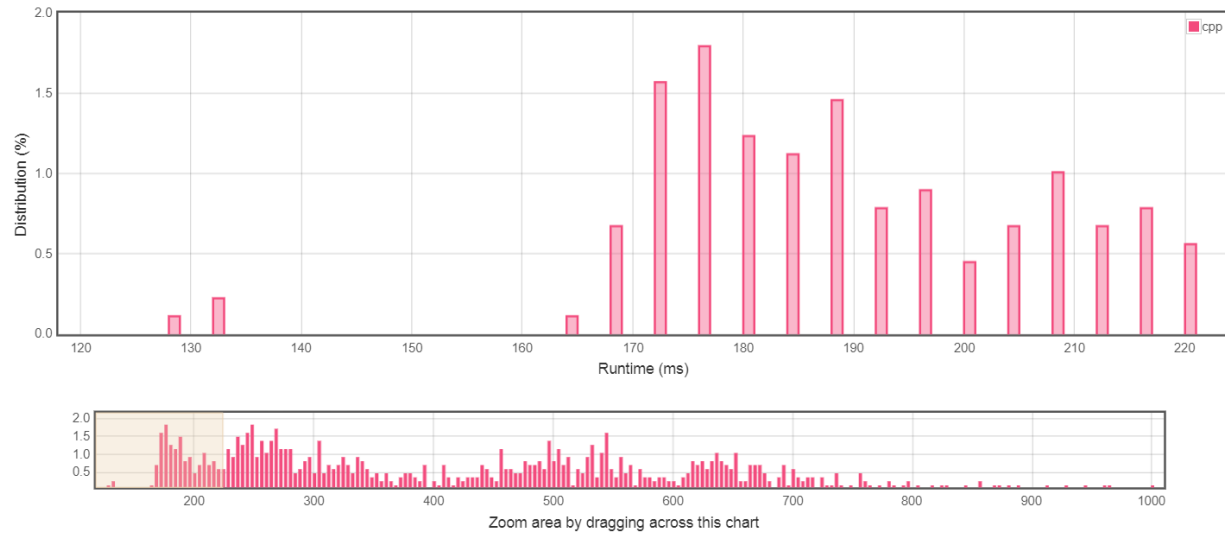
Runtime: 88 ms

Memory Usage: 42.5 MB

Status: Accepted

Submitted: 7 minutes ago

Accepted Solutions Runtime Distribution



Runtime: 88 ms, faster than 100.00% of C++ online submissions for Concatenated Words.

Memory Usage: 42.5 MB, less than 100.00% of C++ online submissions for Concatenated Words.

References