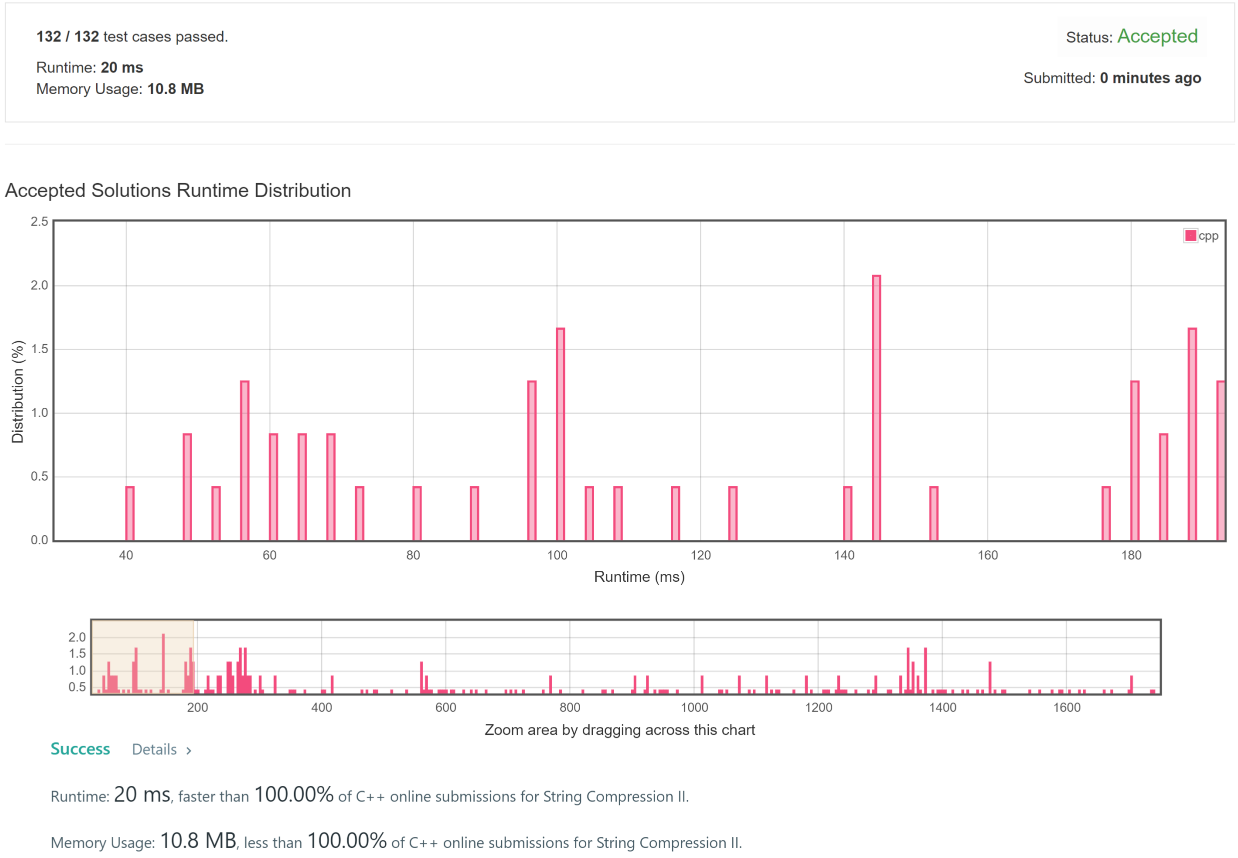
1. Let $f[i][j]$ denote the optimal solution when we compress prefix $1 \sim i$, and delete j characters. We either delete the i-th character c (transit to $f[i-1][j-1]$), or greedily compress a suffix of c's (and delete all intermediate characters other than c). $O(n^2k)$.
2. Notice that when the suffix we compress has length $10^i \sim 10^{i+1} - 1$ ($1 \le i \le \log_{10} n$, special case: $2 \sim 9$ when $i = 0$), the length after compression is the same. Fix the length after compression, we need to support range minimum query in a sliding window. So we can improve the running time by using monotone queues. $O(nk \log n)$.



132 / 132 test cases passed.

Runtime: **20 ms**
Memory Usage: **10.8 MB**

Status: Accepted

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution

Success Details ›

Runtime: 20 ms, faster than 100.00% of C++ online submissions for String Compression II.

Memory Usage: 10.8 MB, less than 100.00% of C++ online submissions for String Compression II.

References