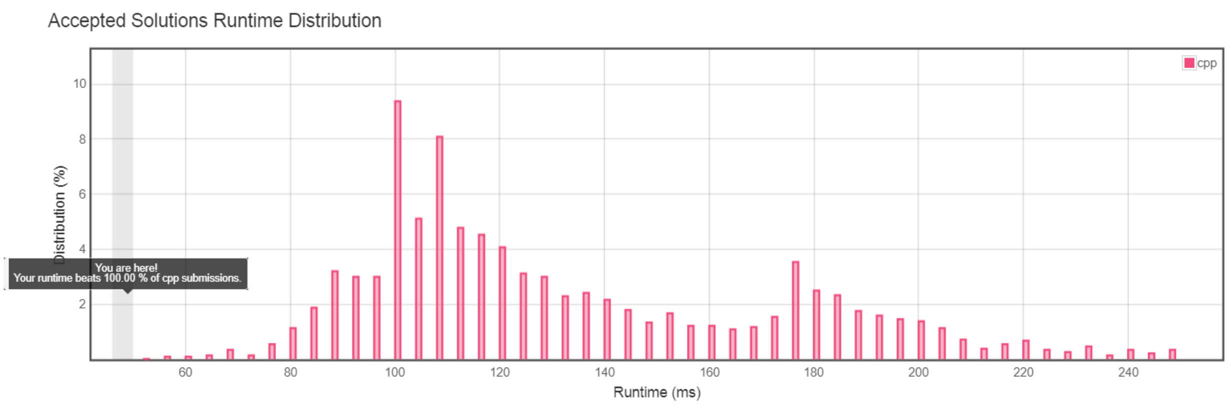
in the worst case, wlog assume all strings have the same length ℓ.
1. for each query, dfs on Trie. $O(n\ell)$ per query.
2. for each query, enumerate all n strings and compare them. speedup this step by word operations. in the worst case, if the answer is no, there are at most $\binom{\ell}{i} \cdot |\Sigma|^i$ strings that differs with the query string in i positions, for all $i \geq 1$ and $\binom{\ell}{i} \cdot |\Sigma|^i \leq n$. assume $\ell >> i$, we approximately have $i \approx \frac{\log n}{\log(\ell|\Sigma|))}$. if we randomly permute the indices in a string globally, in expectation we only need to check $\frac{\ell}{i}$ positions to find the two strings are different, in $O(\frac{\ell \log |\Sigma|}{iw})$ by word operations. in the worst case $\ell = n^{\Theta(1)}$ and $i = O(1)$, in total $O(\frac{n\ell \log |\Sigma|}{w})$ time per query.
3. if we pack the result of $O(\frac{\log n}{\log |\Sigma|})$ consecutive characters into n-bit vectors by method of four russians in $O(n^\epsilon)$ per add, and query by word operations (taking & of $\frac{\ell \log |\Sigma|}{\log n}$ n-bit vectors), we can get $O(\frac{n\ell \log |\Sigma|}{w \log n})$ time per query.

This problem is a generalization of the batch partial matching problem (or orthogonal vectors). For batch partial matching, there are some algorithms that runs slightly better than $O(n^2)$ for small $\ell \approx c \log n$ (e.g. $n^{2-\frac{1}{O(\log c)}}$ time for $\ell = c \log n$ with $c \leq 2^{o(\frac{\log n}{\log \log n})}$ [1, 2]), but for large enough ℓ there's an $\Omega(n^{2-\epsilon})$ lower bound (unless the SETH conjecture is false). see [2] for a survey.

note. we can also get a running time sensitive to the number of ".”s.



References

[1] Amir Abboud, Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 218–230. Society for Industrial and Applied Mathematics, 2015.

[2] Timothy M Chan and Ryan Williams. Deterministic apsp, orthogonal vectors, and more: Quickly derandomizing razborov-smolensky. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1246–1255. Society for Industrial and Applied Mathematics, 2016.