

# A Wordsearch

Given an  $N$  by  $N$  wordsearch and  $N$  words, devise an algorithm to solve the wordsearch in  $O(N^3)$ . For simplicity, assume no word is contained within another, i.e. if the word "bear" is given, "be" wouldn't also be given.

If you are unfamiliar with wordsearches or want to gain some wordsearch solving intuition, see below for an example wordsearch. Note that the below wordsearch doesn't follow the precise specification of an  $N$  by  $N$  wordsearch with  $N$  words, but your algorithm should work on this wordsearch regardless.

## Example Wordsearch:

```
C M U H O S A E D
T R A T H A N K A
O C Y E S R T U T
N I R S A I O L S
Y R R M T N N H R
Y E A E V A R U E
A A A I M E L C R
N H D J Y U A C I
T Y S A A R S U C
A R S I G Y E S A
```

```
ajay      anton
crystal   eric
grace     isha
luke      naama
rica      sarina
sherry    shreyas
sohum     sumer
tony      vidya
```

**Hint:** Add the words to a [Trie](#), and you may find the `longestPrefixOf` operation helpful. Recall that `longestPrefixOf` accepts a `String` key and returns the longest prefix of key that exists in the Trie, or `null` if no prefix exists.