1. Search for given query using standard Trie search algorithm.
2. If query prefix is not present, return –1.
3. If query is present and is end of word in Trie, print query. This can quickly checked by seeing if last matching node has isEndWord flag set. We use this flag in Trie to mark end of word nodes for purpose of searching.
4. If last matching node of query has no children, return.
5. Else recursively print all nodes under subtree of last matching node.

Time complexity

O(n) :

Worst case runtime of creating a trie is O(mn) where m is the longest word and n is the total number of word. The time complexity of **searching**, inserting, and deleting from a trie depends on the length of the word and the total number of the words O(am).

Space Complexity

We don't need extra space so it is o(1)