class TrieNode:

def \_init\_(self):

self.children = {}

self.is\_end = False

class Trie:

def \_init\_(self):

self.root = TrieNode()

def insert(self, word: str) -> None:

node = self.root

for char in word:

if char not in node.children:

node.children[char] = TrieNode()

node = node.children[char]

node.is\_end = True

def search(self, word: str) -> bool:

node = self.root

for char in word:

if char not in node.children:

return False

node = node.children[char]

return node.is\_end

def startsWith(self, prefix: str) -> bool:

node = self.root

for char in prefix:

if char not in node.children:

return False

node = node.children[char]

return True