

词典中最长的单词

给出一个字符串数组 `words` 组成的一本英语词典。从中找出最长的一个单词，该单词是由 `words` 词典中其他单词逐步添加一个字母组成。若其中有多多个可行的答案，则返回答案中字典序最小的单词。

若无答案，则返回空字符串。

示例 1:

输入:

```
words = ["w","wo","wor","worl", "world"]
```

输出: "world"

解释:

单词"world"可由"w", "wo", "wor", 和 "worl"添加一个字母组成。

示例 2:

输入:

```
words = ["a", "banana", "app", "appl", "ap", "apply", "apple"]
```

输出: "apple"

解释:

"apply"和"apple"都能由词典中的单词组成。但是"apple"的字典序小于"apply"。

```
class Solution:
    def longestWord(self, words: List[str]) -> str:
        ...
        按长度升序排序以后，从前往后遍历，检查当前单词除最后一个字母以外，是不是在
        历史记录中，是则在历史记录中加入当前词
        ...
        temp=set([''])#构造空字符的 set 集合
        words=sorted(words,key=len)#按照单词长度 由短到长排序
        for word in words:
            if word[:-1] in temp:
                temp.add(word)
        return max(sorted(temp),key=len)#去重中找到最长的单词返回
```