

## 720. Longest Word in Dictionary

Difficulty: Easy

•Total Accepted:2.7K

•Total Submissions:7.2K

•Contributor:[zestypan](#)

Given a list of strings `words` representing an English Dictionary, find the longest word in `words` that can be built one character at a time by other words in `words`. If there is more than one possible answer, return the longest word with the smallest lexicographical order.

If there is no answer, return the empty string.

### Example 1:

**Input:**

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

**Output:** "world"

**Explanation:**

The word "world" can be built one character at a time by "w", "wo", "wor", and "worl".

### Example 2:

**Input:**

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

**Output:** "apple"

**Explanation:**

Both "apply" and "apple" can be built from other words in the dictionary. However, "apple" is lexicographically smaller than "apply".

### Note:

- All the strings in the input will only contain lowercase letters.
- The length of `words` will be in the range `[1, 1000]`.

- The length of `words[i]` will be in the range `[1, 30]`.

```
#include<stdio.h>
#include<iostream>
#include<vector>
#include<algorithm>
#include<assert.h>
#include<unordered_set>
using namespace std;

bool isIn(string a, string b)
{
    assert(a.size()<b.size());
    for(int i=0;i<(int)(b.size()-a.size()+1;i++)
    {
        string temp = b.substr(i,a.size());
        //cout << temp <<endl;
        if(temp==a)
            return true;
    }
    return false;
}

string longestWord(vector<string>& words) {
    sort(words.begin(),words.end());
    unordered_set<string> res;
    string resword = "";
    for(auto word:words)
    {
        string tempstr = word.substr(0,word.size()-1);
        if(word.size()==1 || res.count(tempstr)>0)
        {
            res.insert(word);
            if(resword==" || word.size()>resword.size())
```

```
        {
            resword = word;
        }
    }
}
return resword;
}
```

```
int main(int argc, char *argv[])
{
    vector<string> words = {"w","wo","wor","worl","world"};
    string ans = longestWord(words);
    cout << ans << endl;
    return 0;
}
```