

## 916. Word Subsets

Medium Topics Companies

You are given two string arrays `words1` and `words2`.

A string `b` is a **subset** of string `a` if every letter in `b` occurs in `a` including multiplicity.

- For example, `"wrr"` is a subset of `"warrior"` but is not a subset of `"world"`.

A string `a` from `words1` is **universal** if for every string `b` in `words2`, `b` is a subset of `a`.

Return an array of all the **universal** strings in `words1`. You may return the answer in **any order**.

### Example 1:

**Input:** `words1 = ["amazon", "apple", "facebook", "google", "leetcode"]`, `words2 = ["e", "o"]`  
**Output:** `["facebook", "google", "leetcode"]`

ith 1: Brute Force

for ...  
for ...

might NOT be efficient enough.

→ hash words 2  
make dic

→ for w in words1.  
hash words 2.  
compare

All Submissions

Accepted 56 / 56 testcases passed  
An... submitted at Jan 10, 2025 10:09

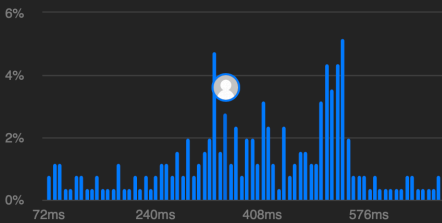
Runtime

353 ms | Beats 64.54%

Analyze Complexity

Memory

21.30 MB | Beats 21.91%



72ms 240ms 408ms 576ms

Python3 Auto

```
1 class Solution:
2     def wordSubsets(self, words1: List[str], words2: List[str]) -> List[str]:
3         dic = defaultdict(int)
4         for w in words2:
5             count_w = Counter(w)
6             for c, n in count_w.items():
7                 dic[c] = max(n, dic[c])
8
9         res = []
10        for w in words1:
11            boo = True
12            count_w = Counter(w)
13            for c, n in dic.items():
14                if count_w[c] < n:
15                    boo = False
16                    break
17
18            if boo:
19                res.append(w)
20
21        return res
```

Ln 14, Col 34 | Saved

Run

Testcase

Test Result