

# Problem 1087: Brace Expansion

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 66.75%

**Paid Only:** Yes

**Tags:** String, Backtracking, Stack, Breadth-First Search, Sorting

## Problem Description

You are given a string `s`` representing a list of words. Each letter in the word has one or more options.

\* If there is one option, the letter is represented as is. \* If there is more than one option, then curly braces delimit the options. For example, ``{a,b,c}`` represents options ``["a", "b", "c"]``.

For example, if `s = "a{b,c}"`, the first character is always ``a``, but the second character can be ``b`` or ``c``. The original list is ``["ab", "ac"]``.

Return all words that can be formed in this manner, **\*\*sorted\*\*** in lexicographical order.

**\*\*Example 1:\*\***

**\*\*Input:\*\*** `s = "{a,b}c{d,e}f"` **\*\*Output:\*\*** `["acdf", "acef", "bcdf", "bcef"]`

**\*\*Example 2:\*\***

**\*\*Input:\*\*** `s = "abcd"` **\*\*Output:\*\*** `["abcd"]`

**\*\*Constraints:\*\***

\* `1 <= s.length <= 50`` \* `s`` consists of curly brackets ``{}``, commas ``,``, and lowercase English letters. \* `s`` is guaranteed to be a valid input. \* There are no nested curly brackets. \* All characters inside a pair of consecutive opening and ending curly brackets are different.

## Code Snippets

### C++:

```
class Solution {  
public:  
    vector<string> expand(string s) {  
  
    }  
};
```

### Java:

```
class Solution {  
    public String[] expand(String s) {  
  
    }  
}
```

### Python3:

```
class Solution:  
    def expand(self, s: str) -> List[str]:
```