

1087. Brace Expansion

A string `S` represents a list of words.

Each letter in the word has 1 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, `"{a,b,c}d{e,f}"` represents the list `["ade", "adf", "bde", "bdf", "cde", "cdf"]`.

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

```
class Solution {
    public String[] expand(String S) {

        // backtracking
        List<String> res = new ArrayList<>();
        dfs(S, 0, new StringBuilder(), res);

        return res.toArray(new String[res.size()]);
    }

    private void dfs(String S, int index, StringBuilder sb, List<String> res)
    {
        if (index == S.length()) {
            if (sb.length() > 0)
                res.add(sb.toString());

            return;
        }

        char c = S.charAt(index);
        int position = sb.length();

        if (c == '{') {
            List<Character> charList = new ArrayList<>();
            int endIndex = index + 1;

            while (endIndex < S.length() && S.charAt(endIndex) != '}') {
                if (Character.isLetter(S.charAt(endIndex)))
                    charList.add(S.charAt(endIndex));

                endIndex++;
            }

            Collections.sort(charList); //??
            for (char cl : charList) {
                sb.append(cl);
                dfs(S, endIndex + 1, sb, res);
                sb.setLength(position);
            }
        } else if (Character.isLetter(c)) {
            sb.append(S.charAt(index));
        }
    }
}
```

```
        dfs(S, index+1, sb, res);  
    }  
}
```