<https://leetcode.com/problems/letter-case-permutation/>

Given a string s, you can transform every letter individually to be lowercase or uppercase to create another string.

Return *a list of all possible strings we could create*. Return the output in **any order**.

**Example 1:**

Input: s = "a1b2"

Output: ["a1b2","a1B2","A1b2","A1B2"]

**Example 2:**

Input: s = "3z4"

Output: ["3z4","3Z4"]

**Constraints:**

* 1 <= s.length <= 12
* s consists of lowercase English letters, uppercase English letters, and digits.

**Attempt 1: 2022-12-22**

**Solution 1: Backtracking (10 min)**

**Style 1: With StringBuilder**

class Solution {

public List<String> letterCasePermutation(String s) {

List<String> result = new ArrayList<String>();

helper(s, 0, result, new StringBuilder());

return result;

}

private void helper(String s, int index, List<String> result, StringBuilder sb) {

if(index == s.length()) {

result.add(sb.toString());

return;

}

int len = sb.length();

char c = s.charAt(index);

if(Character.isLetter(c)) {

helper(s, index + 1, result, sb.append(Character.toUpperCase(c)));

sb.setLength(len);

helper(s, index + 1, result, sb.append(Character.toLowerCase(c)));

sb.setLength(len);

} else {

helper(s, index + 1, result, sb.append(c));

sb.setLength(len);

}

}

}

Time Complexity : O(2^N), because for each character we have two choices, upper case or lower case

Space Complexity : O(2^N)

**Style 2: With String**

class Solution {

public List<String> letterCasePermutation(String s) {

List<String> result = new ArrayList<String>();

helper(s, 0, result, "");

return result;

}

private void helper(String s, int index, List<String> result, String str) {

if(index == s.length()) {

result.add(str);

return;

}

char c = s.charAt(index);

if(Character.isLetter(c)) {

helper(s, index + 1, result, str + Character.toUpperCase(c));

helper(s, index + 1, result, str + Character.toLowerCase(c));

} else {

helper(s, index + 1, result, str + c);

}

}

}

Time Complexity : O(2^N), because for each character we have two choices, upper case or lower case

Space Complexity : O(2^N)