<https://zhengyang2015.gitbooks.io/lintcode/content/string_permutation_ii_10.html>

## **Question:**

Given a string, find all permutations of it without duplicates.

Example

Given "abb", return ["abb", "bab", "bba"].

Given "aabb", return ["aabb", "abab", "baba", "bbaa", "abba", "baab"].

**Attempt 1: 2022-12-27**

**Solution 1: Backtracking (10 min)**

public class Solution {

public List<String> stringPermutation2(String str) {

char[] chars = str.toCharArray();

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

boolean[] visited = new boolean[chars.length];

Arrays.sort(chars);

helper(chars, result, visited, new StringBuilder());

return result;

}

private void helper(char[] chars, List<String> result, boolean[] visited, StringBuilder sb) {

if(sb.length() == chars.length) {

result.add(sb.toString());

return;

}

for(int i = 0; i < chars.length; i++) {

if(visited[i] || (i > 0 && !visited[i - 1] && chars[i] == chars[i - 1])) {

continue;

}

visited[i] = true;

sb.append(chars[i]);

helper(chars, result, visited, sb);

visited[i] = false;

sb.deleteCharAt(sb.length() - 1);

}

}

}

**Refer to**

<https://zhengyang2015.gitbooks.io/lintcode/content/string_permutation_ii_10.html>

public class Solution {

/\*\*

\* @param str a string

\* @return all permutations

\*/

public List<String> stringPermutation2(String str) {

// Write your code here

char[] string = str.toCharArray();

boolean[] isUsed = new boolean[string.length];

Arrays.sort(string);

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

String temp = new String();

stringPermutation2Helper(result, temp, string, isUsed);

return result;

}

private void stringPermutation2Helper(List<String> result,

String temp,

char[] string,

boolean[] isUsed) {

if (temp.length() == string.length) {

result.add(temp);

return;

}

for (int i = 0; i < string.length; i++) {

if (isUsed[i] == true || i != 0 &&

isUsed[i - 1] == false &&

string[i] == string[i - 1]) {

continue;

}

isUsed[i] = true;

stringPermutation2Helper(result, temp + string[i], string, isUsed);

isUsed[i] = false;

}

}

}