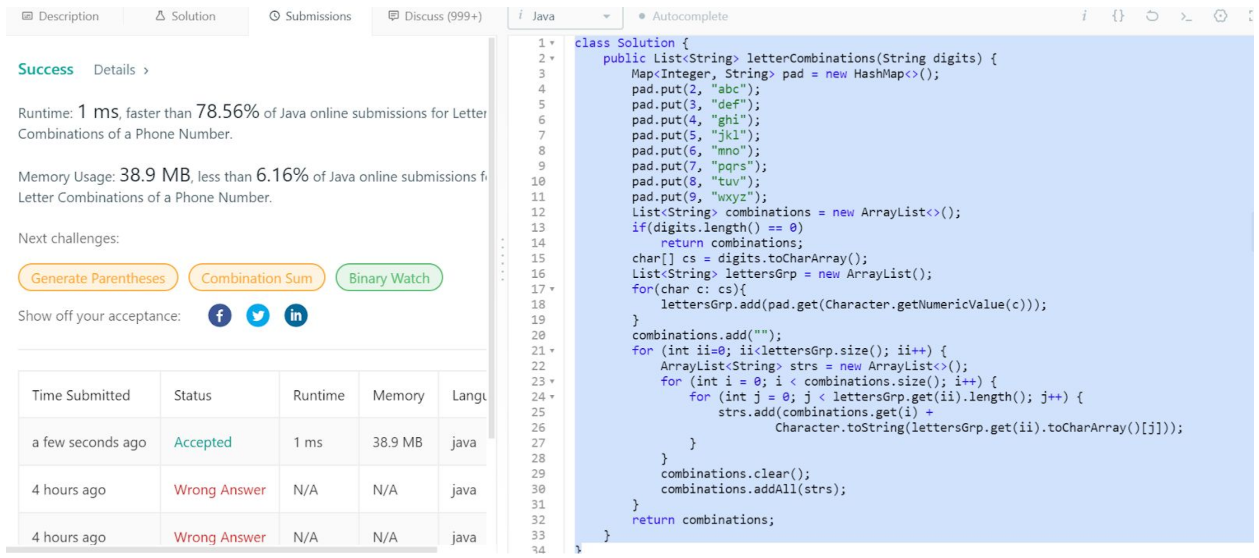
Problem: **Letter Combinations of a Phone Number**

Student: Ovidiu Mura



Success Details ›

Runtime: 1 ms, faster than 78.56% of Java online submissions for Letter Combinations of a Phone Number.

Memory Usage: 38.9 MB, less than 6.16% of Java online submissions fo Letter Combinations of a Phone Number.

Next challenges:

Generate Parentheses Combination Sum Binary Watch

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Langu
a few seconds ago	Accepted	1 ms	38.9 MB	java
4 hours ago	Wrong Answer	N/A	N/A	java
4 hours ago	Wrong Answer	N/A	N/A	java

```java
class Solution {
    public List<String> letterCombinations(String digits) {
        Map<Integer, String> pad = new HashMap<>();
        pad.put(2, "abc");
        pad.put(3, "def");
        pad.put(4, "ghi");
        pad.put(5, "jkl");
        pad.put(6, "mno");
        pad.put(7, "pqrs");
        pad.put(8, "tuv");
        pad.put(9, "wxyz");
        List<String> combinations = new ArrayList<>();
        if(digits.length() == 0)
            return combinations;
        char[] cs = digits.toCharArray();
        List<String> lettersGrp = new ArrayList();
        for(char c: cs){
            lettersGrp.add(pad.get(Character.getNumericValue(c)));
        }
        combinations.add("");
        for (int ii=0; ii<lettersGrp.size(); ii++) {
            ArrayList<String> strs = new ArrayList<>();
            for (int i = 0; i < combinations.size(); i++) {
                for (int j = 0; j < lettersGrp.get(ii).length(); j++) {
                    strs.add(combinations.get(i) +
                            Character.toString(lettersGrp.get(ii).toCharArray()[j]));
                }
            }
            combinations.clear();
            combinations.addAll(strs);
        }
        return combinations;
    }
}
```

```java
class Solution {
    public List<String> letterCombinations(String digits) {
        Map<Integer, String> pad = new HashMap<>();
        pad.put(2, "abc");
        pad.put(3, "def");
        pad.put(4, "ghi");
        pad.put(5, "jkl");
        pad.put(6, "mno");
        pad.put(7, "pqrs");
        pad.put(8, "tuv");
        pad.put(9, "wxyz");
        List<String> combinations = new ArrayList<>();
        if(digits.length() == 0)
            return combinations;
        char[] cs = digits.toCharArray();
        List<String> lettersGrp = new ArrayList();
        for(char c: cs){
            lettersGrp.add(pad.get(Character.getNumericValue(c)));
        }
        combinations.add("");
        for (int ii=0; ii<lettersGrp.size(); ii++) {
            ArrayList<String> strs = new ArrayList<>();
            for (int i = 0; i < combinations.size(); i++) {
                for (int j = 0; j < lettersGrp.get(ii).length(); j++) {
                    strs.add(combinations.get(i) +
                            Character.toString(lettersGrp.get(ii).toCharArray()[j]));
                }
            }
            combinations.clear();
            combinations.addAll(strs);
        }
        return combinations;
    }
}
```