ASSIGNMENT - 4

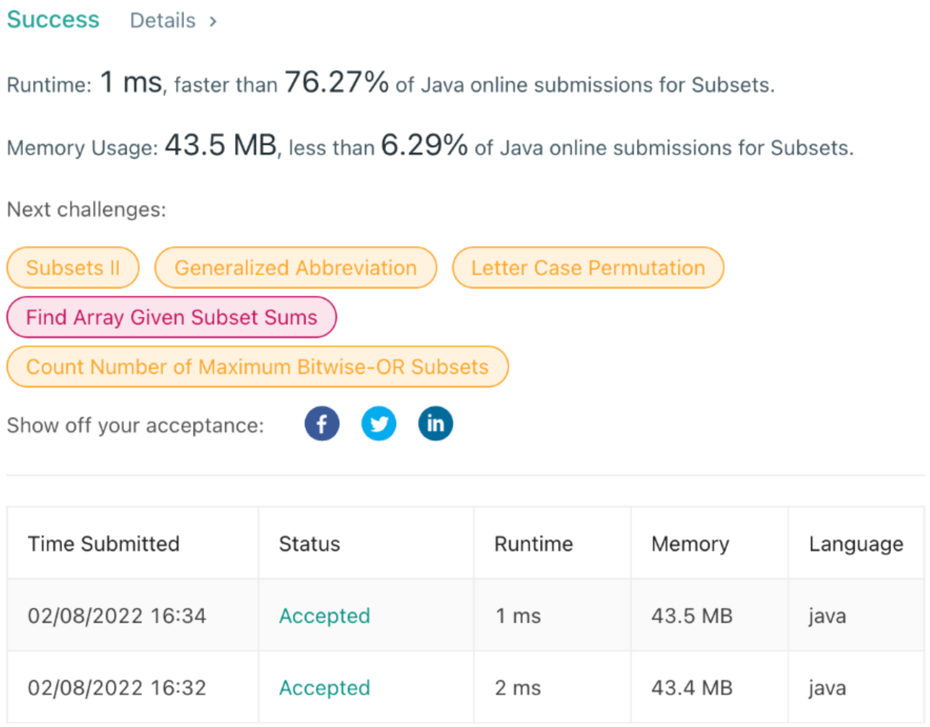
1. Solve the leetcode question no. 78 (Subsets).
Implement a solution that can be accepted.
Provide a screenshot of your submission. Also post your source code here.

Solution:

```
class Solution {
    public List<List<Integer>> subsets(int[] nums) {
        Arrays.sort(nums);
        List<List<Integer>> result = new ArrayList<>();
        result.add(new ArrayList<>());

        for (int i = 0; i < nums.length; ++i) {
            int size = result.size();
            for (int j = 0; j < size; ++j) {
                List<Integer> subset = new ArrayList<>(result.get(j));
                subset.add(nums[i]);
                result.add(subset);
            }
        }
        return result;
    }
}
```

Screenshot:



Success   Details ›

Runtime: 1 ms, faster than 76.27% of Java online submissions for Subsets.

Memory Usage: 43.5 MB, less than 6.29% of Java online submissions for Subsets.

Next challenges:

Subsets II   Generalized Abbreviation   Letter Case Permutation
Find Array Given Subset Sums
Count Number of Maximum Bitwise-OR Subsets

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
02/08/2022 16:34	Accepted	1 ms	43.5 MB	java
02/08/2022 16:32	Accepted	2 ms	43.4 MB	java

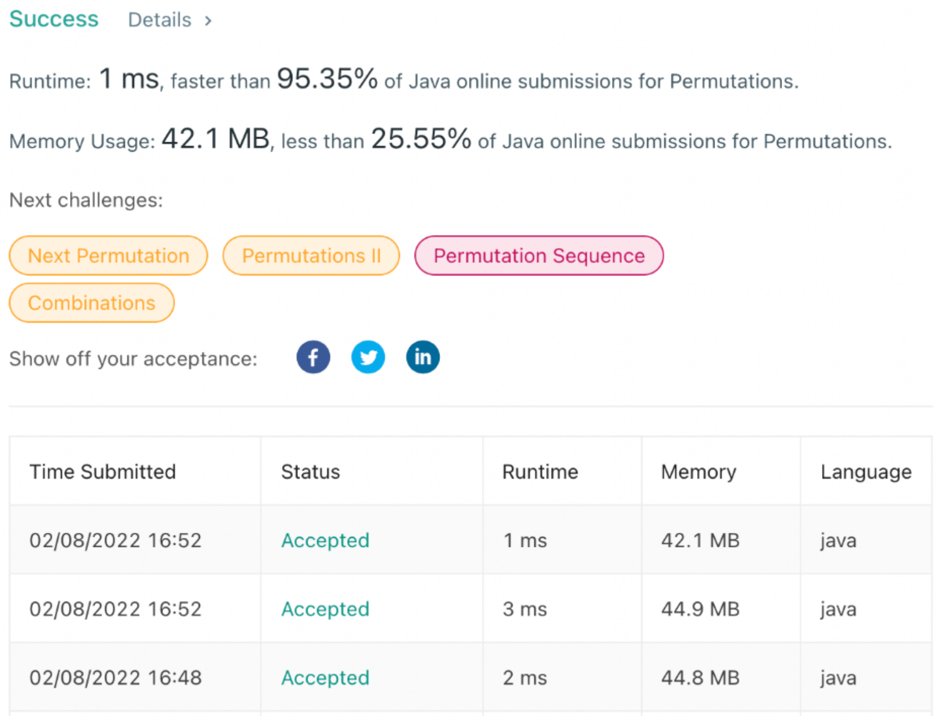
2. Solve the leetcode question no. 46 (Permutations)
Implement a solution that can be accepted.
Provide a screenshot of your submission. Also post your source code here.

Solution:

```
class Solution {
    public List<List<Integer>> permute(int[] nums) {
        List<List<Integer>> list = new ArrayList<>();
        backtrack(list, new ArrayList<>(), nums);
        return list;
    }

    private void backtrack(List<List<Integer>> list, List<Integer> temp, int [] nums) {
        if(temp.size() == nums.length) {
            list.add(new ArrayList<>(temp));
        } else {
            for(int i = 0; i < nums.length; i++) {
                if(temp.contains(nums[i]))
                    continue;
                temp.add(nums[i]);
                backtrack(list, temp, nums);
                temp.remove(temp.size() - 1);
            }
        }
    }
}
```

Screenshot:



Runtime: 1 ms, faster than 95.35% of Java online submissions for Permutations.

Memory Usage: 42.1 MB, less than 25.55% of Java online submissions for Permutations.

Next challenges:

Time Submitted	Status	Runtime	Memory	Language
02/08/2022 16:52	Accepted	1 ms	42.1 MB	java
02/08/2022 16:52	Accepted	3 ms	44.9 MB	java
02/08/2022 16:48	Accepted	2 ms	44.8 MB	java