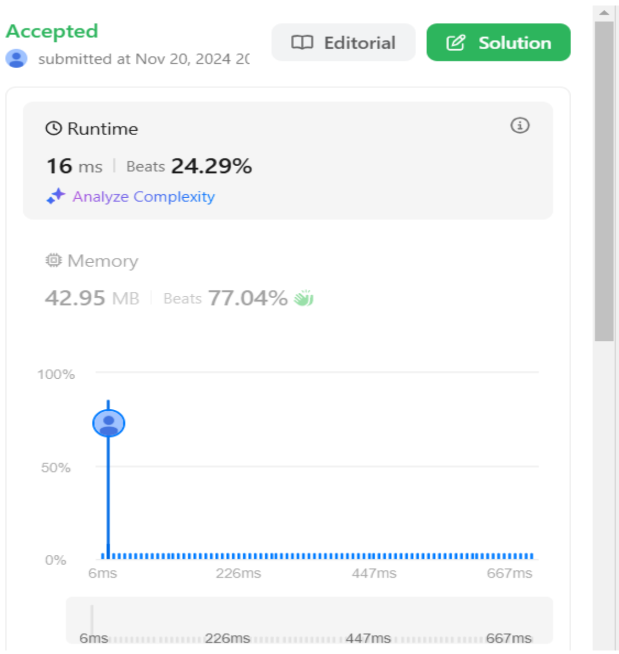
DSA PRACTICE PROBLEMS

20/11/2024

1. 3 Sum closest: TC: O(N^2)



Accepted

submitted at Nov 20, 2024 2(

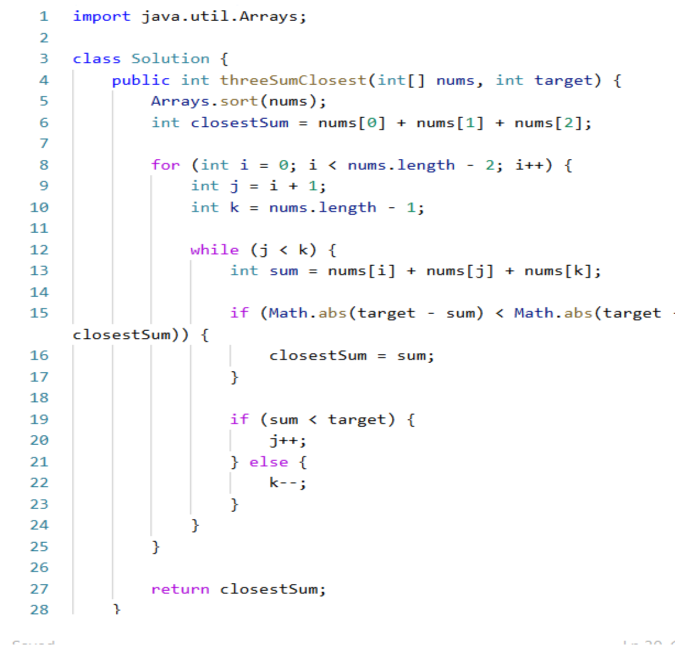
Runtime

16 ms | Beats 24.29%

Analyze Complexity

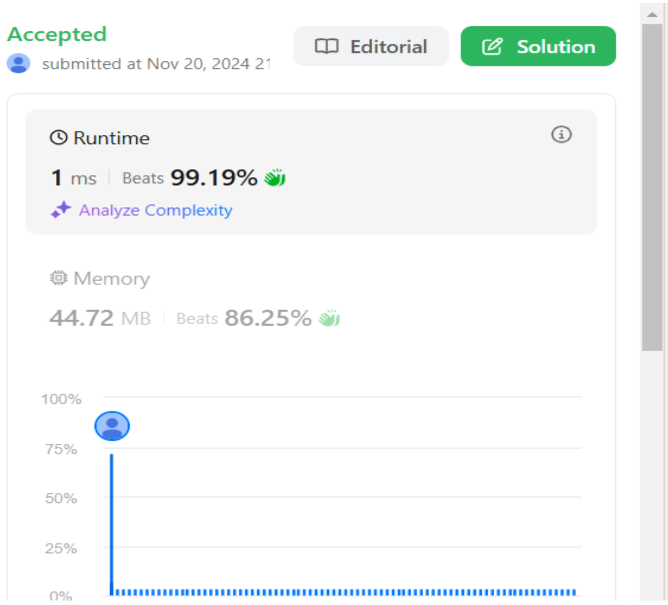
Memory

42.95 MB | Beats 77.04%



```
import java.util.Arrays;

class Solution {
    public int threeSumClosest(int[] nums, int target) {
        Arrays.sort(nums);
        int closestSum = nums[0] + nums[1] + nums[2];

        for (int i = 0; i < nums.length - 2; i++) {
            int j = i + 1;
            int k = nums.length - 1;

            while (j < k) {
                int sum = nums[i] + nums[j] + nums[k];

                if (Math.abs(target - sum) < Math.abs(target -
closestSum)) {
                    closestSum = sum;
                }

                if (sum < target) {
                    j++;
                } else {
                    k--;
                }
            }
        }

        return closestSum;
    }
```

2.JUMP GAME II:



Accepted

submitted at Nov 20, 2024 21

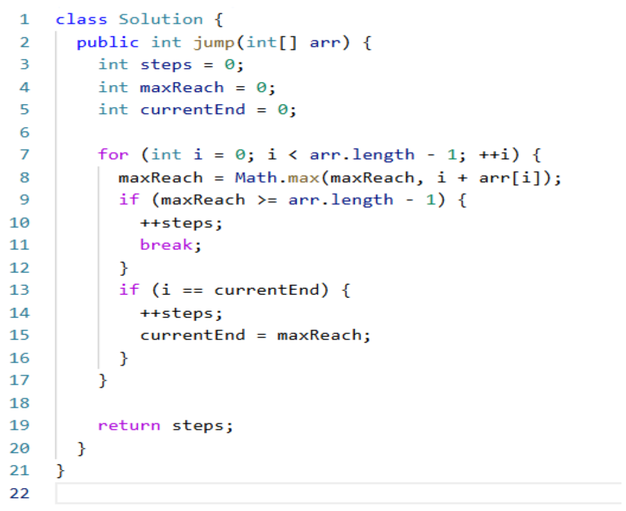
Runtime

1 ms | Beats 99.19%

Analyze Complexity

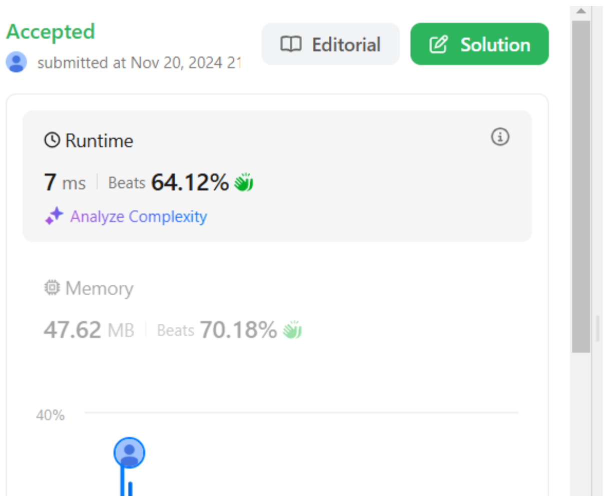
Memory

44.72 MB | Beats 86.25%



```
class Solution {
  public int jump(int[] arr) {
    int steps = 0;
    int maxReach = 0;
    int currentEnd = 0;

    for (int i = 0; i < arr.length - 1; ++i) {
      maxReach = Math.max(maxReach, i + arr[i]);
      if (maxReach >= arr.length - 1) {
        ++steps;
        break;
      }
      if (i == currentEnd) {
        ++steps;
        currentEnd = maxReach;
      }
    }

    return steps;
  }
}
```

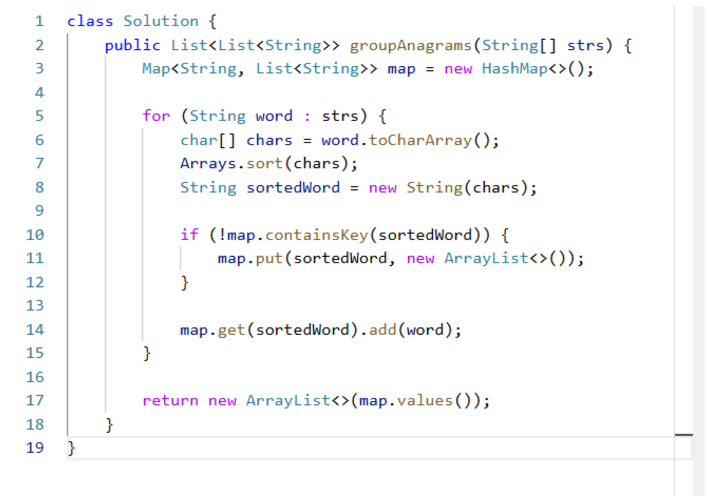
3.GROUP ANAGRAM:



Accepted
submitted at Nov 20, 2024 21

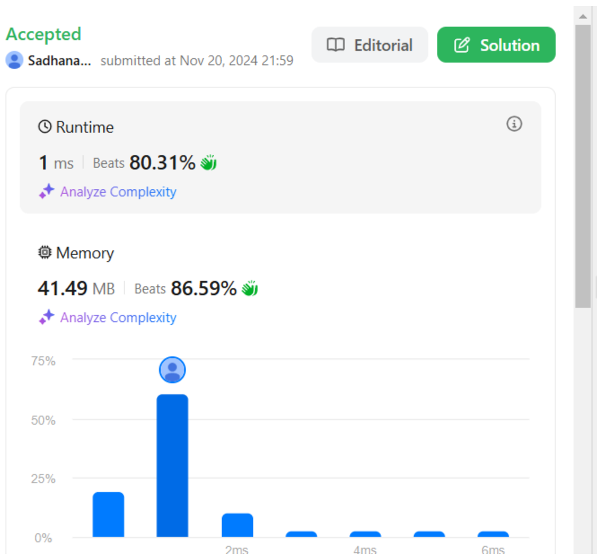
Runtime
7 ms | Beats 64.12%

Memory
47.62 MB | Beats 70.18%



```java
class Solution {
    public List<List<String>> groupAnagrams(String[] strs) {
        Map<String, List<String>> map = new HashMap<>();

        for (String word : strs) {
            char[] chars = word.toCharArray();
            Arrays.sort(chars);
            String sortedWord = new String(chars);

            if (!map.containsKey(sortedWord)) {
                map.put(sortedWord, new ArrayList<>());
            }

            map.get(sortedWord).add(word);
        }

        return new ArrayList<>(map.values());
    }
}
```

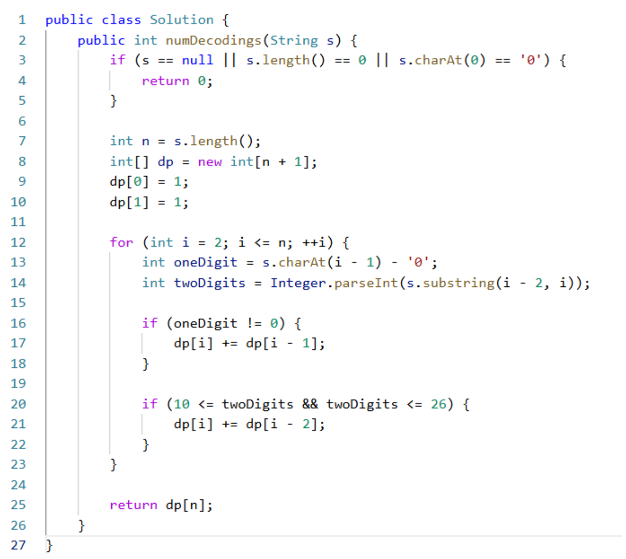
4.DECODE WAYS :



Accepted
Sadhana... submitted at Nov 20, 2024 21:59

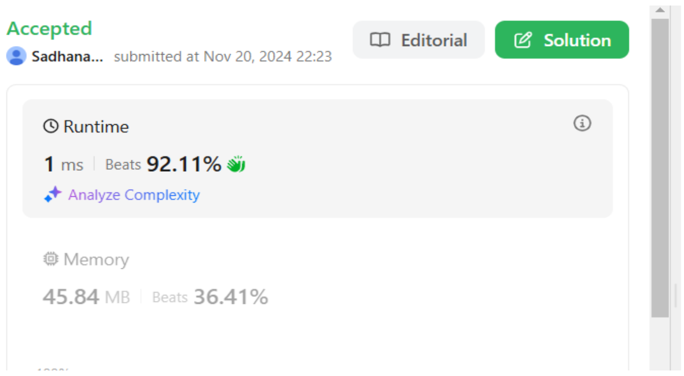
Runtime
1 ms | Beats 80.31%

Memory
41.49 MB | Beats 86.59%



```java
public class Solution {
    public int numDecodings(String s) {
        if (s == null || s.length() == 0 || s.charAt(0) == '0') {
            return 0;
        }

        int n = s.length();
        int[] dp = new int[n + 1];
        dp[0] = 1;
        dp[1] = 1;

        for (int i = 2; i <= n; ++i) {
            int oneDigit = s.charAt(i - 1) - '0';
            int twoDigits = Integer.parseInt(s.substring(i - 2, i));

            if (oneDigit != 0) {
                dp[i] += dp[i - 1];
            }

            if (10 <= twoDigits && twoDigits <= 26) {
                dp[i] += dp[i - 2];
            }
        }

        return dp[n];
    }
}
```

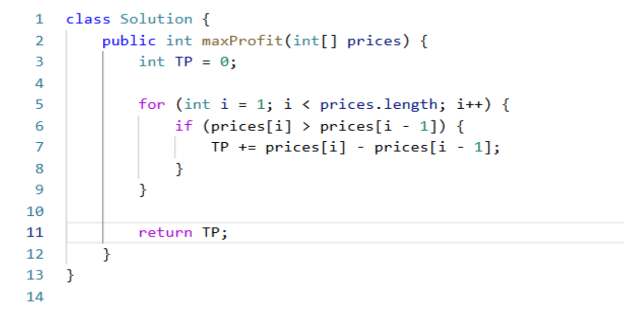
5. Best Time to Buy and Sell Stock II:



Accepted
Sadhana... submitted at Nov 20, 2024 22:23

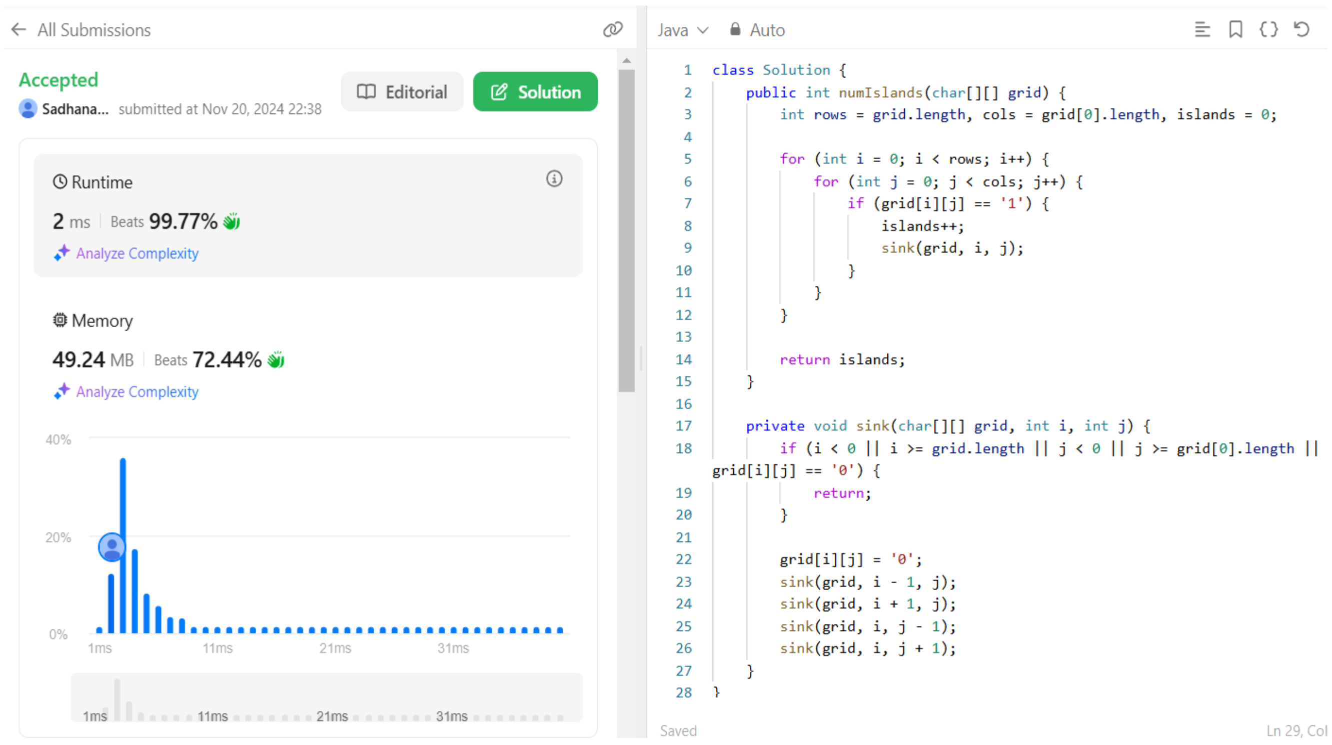
Runtime
1 ms | Beats 92.11%

Memory
45.84 MB | Beats 36.41%



```java
class Solution {
    public int maxProfit(int[] prices) {
        int TP = 0;

        for (int i = 1; i < prices.length; i++) {
            if (prices[i] > prices[i - 1]) {
                TP += prices[i] - prices[i - 1];
            }
        }

        return TP;
    }
}

```

6.NUMBER OF ISLANDS:



```java
class Solution {
    public int numIslands(char[][] grid) {
        int rows = grid.length, cols = grid[0].length, islands = 0;

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                if (grid[i][j] == '1') {
                    islands++;
                    sink(grid, i, j);
                }
            }
        }

        return islands;
    }

    private void sink(char[][] grid, int i, int j) {
        if (i < 0 || i >= grid.length || j < 0 || j >= grid[0].length ||
grid[i][j] == '0') {
            return;
        }

        grid[i][j] = '0';
        sink(grid, i - 1, j);
        sink(grid, i + 1, j);
        sink(grid, i, j - 1);
        sink(grid, i, j + 1);
    }
}
```