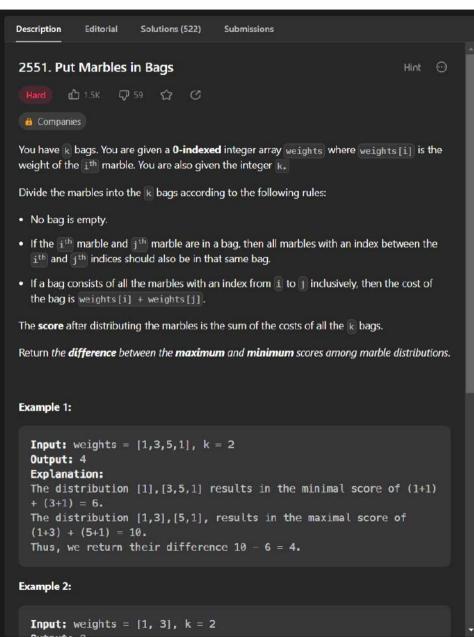


```
i Java V - Auto
         public List<List<Integer>> permuteUnique(int[] nums) {
             List<List<Integer>> matrix = new ArrayList<>();
             List<Integer> row = new ArrayList<>();
             List<Integer> list = new ArrayList<>();
             for (int num : nums) {
                 list.add(num);
             permutelist(list, row, matrix);
             return matrix;
         public static void permutelist(List<Integer> list, List<Integer> row, List<List<Integer>> matrix)
             if (list.size() == 0) {
                 if(matrix.contains(row)){
                     return;
                 matrix.add(new ArrayList<>(row));
Testcase
         Result
Accepted Runtime: 0 ms
  • Case 1
              Case 2
Input
  [1,1,2]
Output
  [[1,1,2],[1,2,1],[2,1,1]]
Expected
Console Y
                                                                                                   Submit
                                                                                          Run
```



```
i Java V - Auto
         public long putMarbles(int[] weights, int k) {
              int n = weights.length;
             int[] pairs = new int[n - 1];
             for (int i = 1; i < n; i \leftrightarrow) {
                  pairs[i - 1] = weights[i] + weights[i - 1];
             Arrays.sort(pairs);
              long minScore = 0;
              long maxScore = 0;
              for (int i = 0; i < k - 1; i++) {
                 minScore += pairs[i];
                  maxScore += pairs[n - i - 2];
              return maxScore - minScore;
 17 }
Testcase
         Result
Accepted Runtime: 0 ms
  Case 1
              · Case 2
Input
  [1,3,5,1]
  2
Output
                                                                                                       Submit
Console Y
                                                                                             Run
```