3Sum

Given an array S of n integers, are there elements a, b, c in S such that a + b + c = 0? Find all unique triplets in the array which gives the sum of zero.

Note: The solution set must not contain duplicate triplets.

For example, given array S = [-1, 0, 1, 2, -1, -4],

A solution set is:

[ [-1, 0, 1], [-1, -1, 2] ]

**T- Talk (/Listen/Clarify)**

**E-Examples (/Test/TDD, Out of the Box)**

-Empty array: [] -> throws an exception

-Less than 3: [-1] -> throws an exception

-None add up to 0: [1, 2, 3] -> empty list

-Possible solution and duplicate numbers [-1, -1, 0, 1] -> [1, 0, 1]

-Answer with duplicate elements: [2, 2, -4, 5] -> [2, 2, -4]

**B- Brute Force**

-Generate every combination of size 3 for the array and check if when added equals to 0. If it does add it to a list of arrays and at the end print the list of arrays

**O- Optimize**

-Similar to the two sum problem

-Store all the values from the array in a hashset

-Generate every combination of size 2 and store them in arrays and calculate their sum

-Map each array to its sum

-Iterate over the hashmap to see if any value would make the sum equal to 0 and print the 3 numbers

**W- Walk Through**

-Having trouble summing each array in the arraylist so it just prints all of the combinations of size 3 of the array as of now

**I- Implement**

-only was able to partially implement the brute force so far

public static void main(String[] args) {

int arr[] = {1, 2, 3, 4};

int size = 3;

int n = arr.length;

Print(arr, n, size);

}

static void MakeCombinations(int arr[], int temp[], int start, int end, int index, int size) {

ArrayList<int[]> allCombinations = new ArrayList<int[]>();

int[] threeElements = new int[3];

if (index == size) {

for (int i = 0; i<size; i++) {

System.out.print(temp[i] + " ");

threeElements[i] = temp[i];

}

allCombinations.add(threeElements);

System.out.println("");

return;

}

for (int i = start; i <= end && end - i + 1 >= size - index; i++) {

temp[index] = arr[i];

MakeCombinations(arr, temp, i+1, end, index+1, size);

}

}

static void Print(int arr[], int n, int size) {

int temp[] = new int[size];

MakeCombinations(arr, temp, 0, n-1, 0, size);

}

**T- Test**

-Having trouble summing each array in the arraylist so it just prints all of the combinations of size 3 of the array as of now