

splitArrayHarder: Java Implementation

call by value

only modified in base case

```
public ArrayList<ArrayList<Integer>> splitArrayHarder(int[] ns) {  
    ArrayList<ArrayList<Integer>> output = new ArrayList<>();  
    splitArrayHarderHelper(ns, 0, 0, 0, new ArrayList<Integer>(), new ArrayList<Integer>(), output);  
    if(output.isEmpty()) {  
        output.add(new ArrayList<Integer>());  
        output.add(new ArrayList<Integer>());  
    }  
    return output;  
}
```

g1 g2

either of them modified in every recursive call (non-base case)

```
private void splitArrayHarderHelper(int[] ns, int i, int sumOfGroup1, int sumOfGroup2,  
    ArrayList<Integer> group1, ArrayList<Integer> group2, ArrayList<ArrayList<Integer>> output) {  
    if(i == ns.length) {  
        if(sumOfGroup1 == sumOfGroup2) {  
            output.add(group1);  
            output.add(group2);  
        }  
    }  
    else {  
        ArrayList<Integer> group1Extended = new ArrayList<>(group1);  
        group1Extended.add(ns[i]);  
        splitArrayHarderHelper(ns, i + 1, sumOfGroup1 + ns[i], sumOfGroup2, group1Extended, group2, output);  
  
        if(output.isEmpty()) {  
            ArrayList<Integer> group2Extended = new ArrayList<>(group2);  
            group2Extended.add(ns[i]);  
            splitArrayHarderHelper(ns, i + 1, sumOfGroup1, sumOfGroup2 + ns[i], group1, group2Extended, output);  
        }  
    }  
}
```

splitArrayHarder: Tracing

Exercise: Trace in Eclipse

@Test

```
public void testSplitArrayHarder_03() {  
    RecursiveMethods rm = new RecursiveMethods();  
    int[] input = {5, 2, 3};  
    ArrayList<ArrayList<Integer>> output = rm.splitArrayHarder(input);  
    ArrayList<Integer> expectedGroup1 = new ArrayList<>(Arrays.asList(5));  
    ArrayList<Integer> expectedGroup2 = new ArrayList<>(Arrays.asList(2, 3));  
    assertTrue(output.size() == 2);  
    assertEquals(expectedGroup1, output.get(0));  
    assertEquals(expectedGroup2, output.get(1));  
}
```

output: $\langle \langle 5 \rangle, \langle 2, 3 \rangle \rangle$

