

# Dry run duplicates algorithm



```
public static void findDuplicatesWithoutHashSet(int[] arr) {
    for (int i = 0; i < arr.length; i++) {
        for (int j = i + 1; j < arr.length; j++) {
            if (arr[i] == arr[j]) {
                System.out.println("Duplicate: " + arr[i]);
                break; // Break inner loop after finding one duplicate
            }
        }
    }
}

// Dry run for int[] arr = {1, 2, 3, 5, 1, 2, 3};
```

Let's dry run the given function for `int[] arr = {1, 2, 3, 5, 1, 2, 3};`.

## Code:

```
java

public static void findDuplicatesWithoutHashSet(int[] arr) {
    for (int i = 0; i < arr.length; i++) {
        for (int j = i + 1; j < arr.length; j++) {
            if (arr[i] == arr[j]) {
                System.out.println("Duplicate: " + arr[i]);
                break; // Break inner loop after finding one duplicate
            }
        }
    }
}
```

## Dry Run:

### Given array:

`arr = {1, 2, 3, 5, 1, 2, 3}`

**Outer loop (i)** runs from `0` to `arr.length - 1`, and **inner loop (j)** runs from `i + 1` to `arr.length - 1`.

### Step-by-step Execution:

i	j	arr[i]	arr[j]	Condition ( <code>arr[i] == arr[j]</code> )?	Output	Action
0	1	1	2	No	-	Continue
0	2	1	3	No	-	Continue
0	3	1	5	No	-	Continue
0	4	1	1	Yes	"Duplicate: 1"	Break inner loop
1	2	2	3	No	-	Continue

i	j	arr[i]	arr[j]	Condition (`arr[i] == arr[j]`)?	Output	Action
1	3	2	5	No	-	Continue
1	4	2	1	No	-	Continue
1	5	2	2	Yes	"Duplicate: 2"	Break inner loop
2	3	3	5	No	-	Continue
2	4	3	1	No	-	Continue
2	5	3	2	No	-	Continue
2	6	3	3	Yes	"Duplicate: 3"	Break inner loop
3	4	5	1	No	-	Continue
3	5	5	2	No	-	Continue
3	6	5	3	No	-	Continue
4	5	1	2	No	-	Continue
4	6	1	3	No	-	Continue
5	6	2	3	No	-	Continue

## Final Output:

```
makefile
```

```
Duplicate: 1
Duplicate: 2
Duplicate: 3
```

## Explanation:

- The function iterates through each element and checks for duplicates using a nested loop.
- When a duplicate is found, it prints the value and **breaks** the inner loop (to avoid printing multiple duplicates of the same element).
- This ensures that each duplicate is reported only **once**, even if it appears more than twice in the array.

Would you like an optimized version of this? 🚀