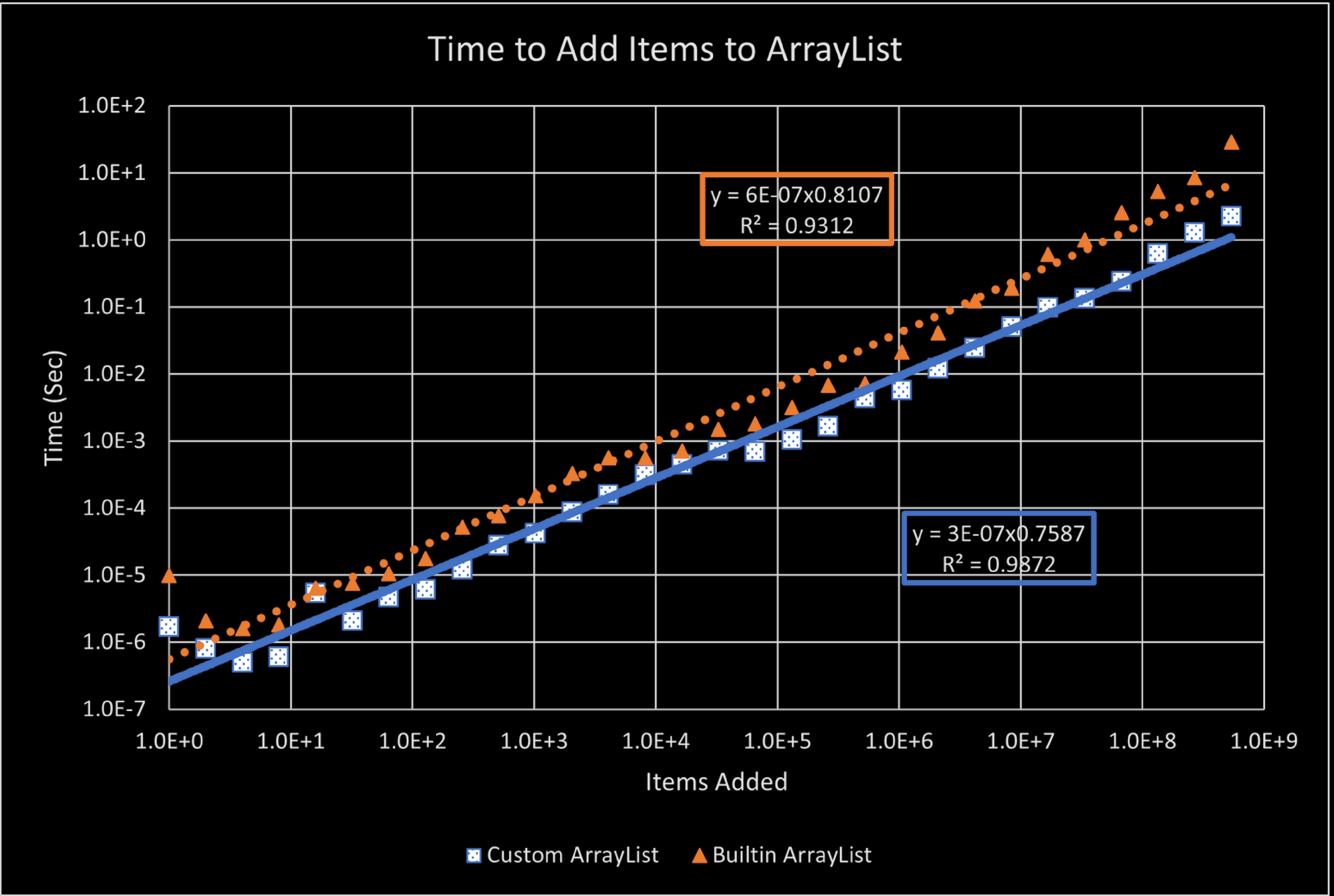
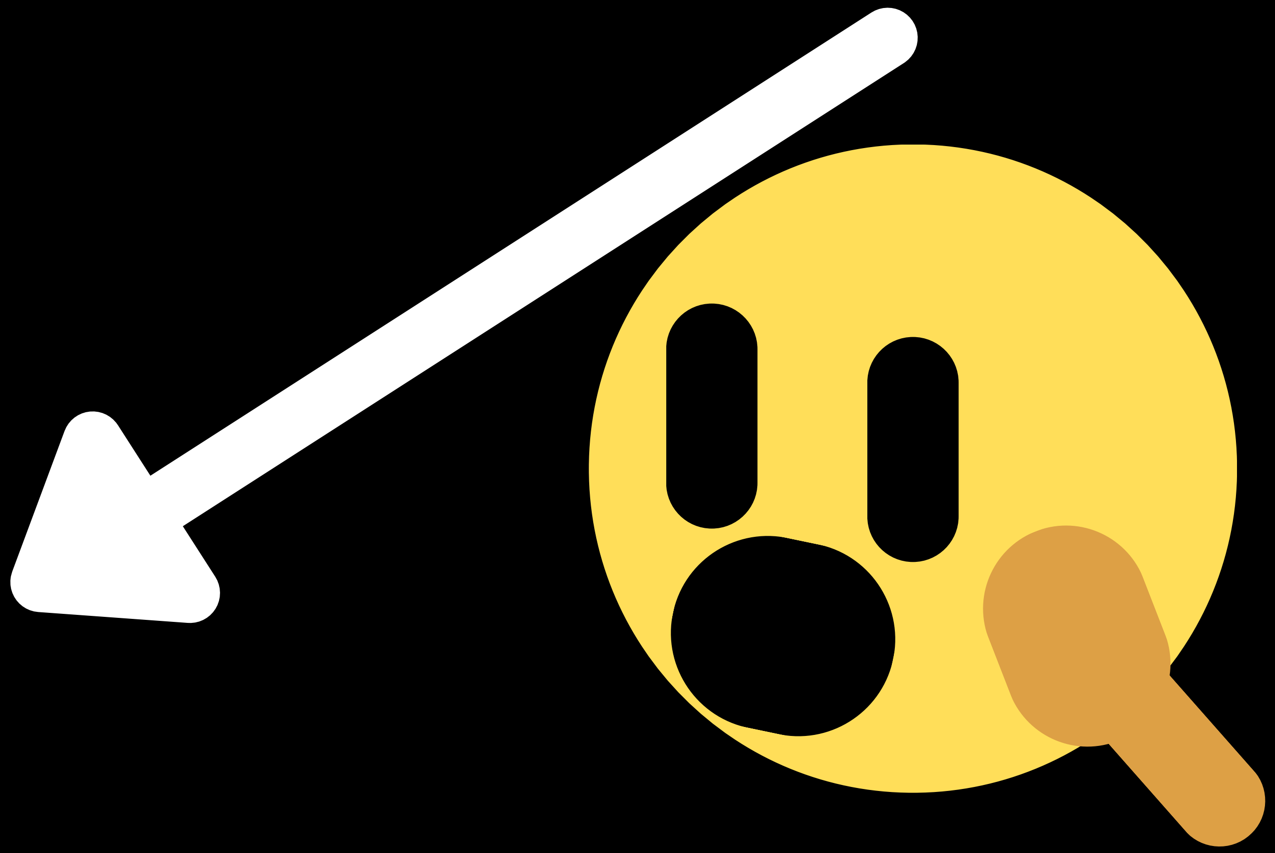
I made a custom ArrayList

wooooooooooooooooooooooooooooooooooo

Hey look, mine's slightly faster!



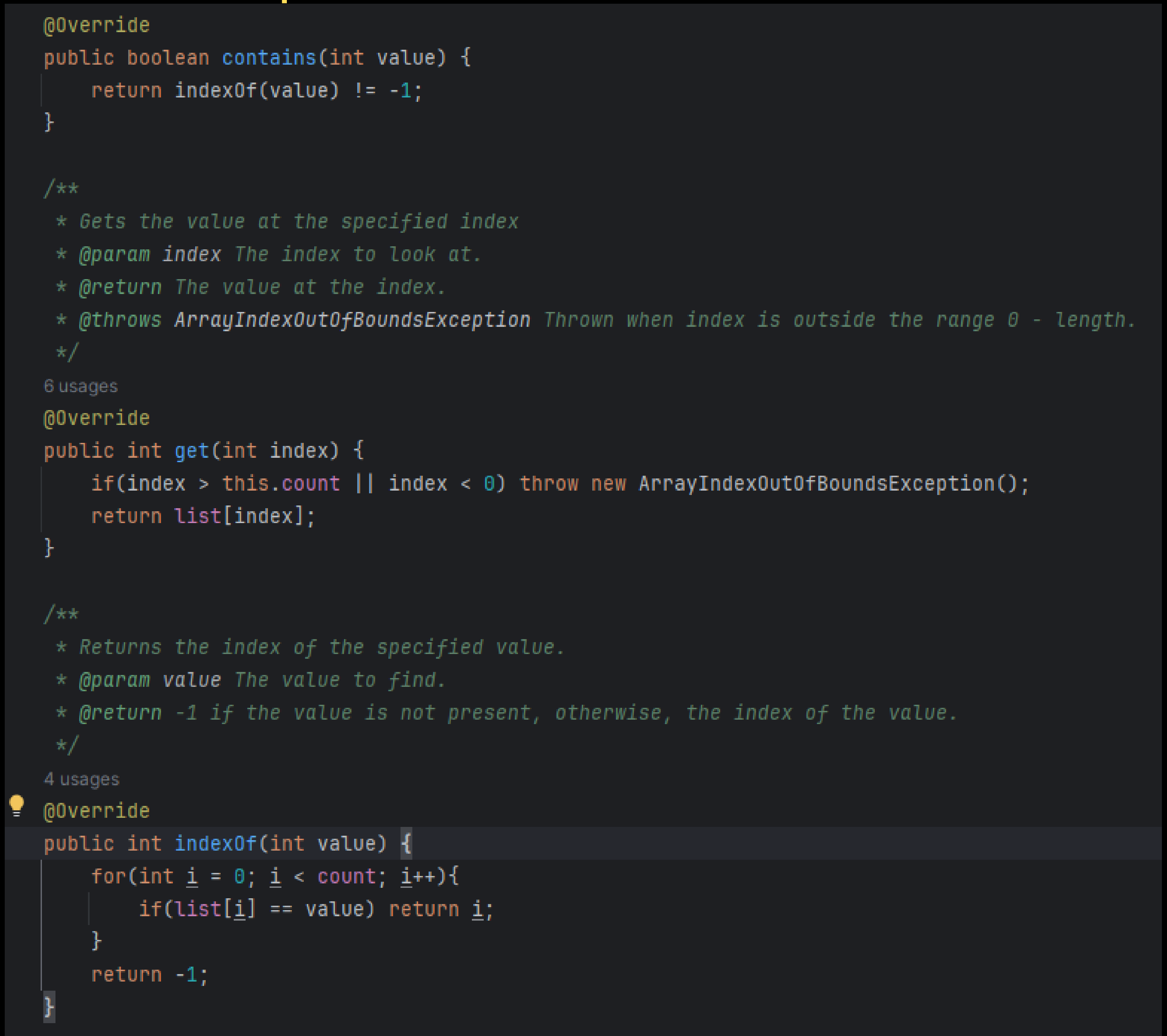


Somehow, the Custom add function is slightly faster on average. Not only that, but if the trendline is to be believed, it's going to become faster and faster.

It does everything it’s supposed to.

- Equals
- Add
- Add All
- Clear
- Contains
- Get
- Index Of
- Is Empty
- Pop

For Example:



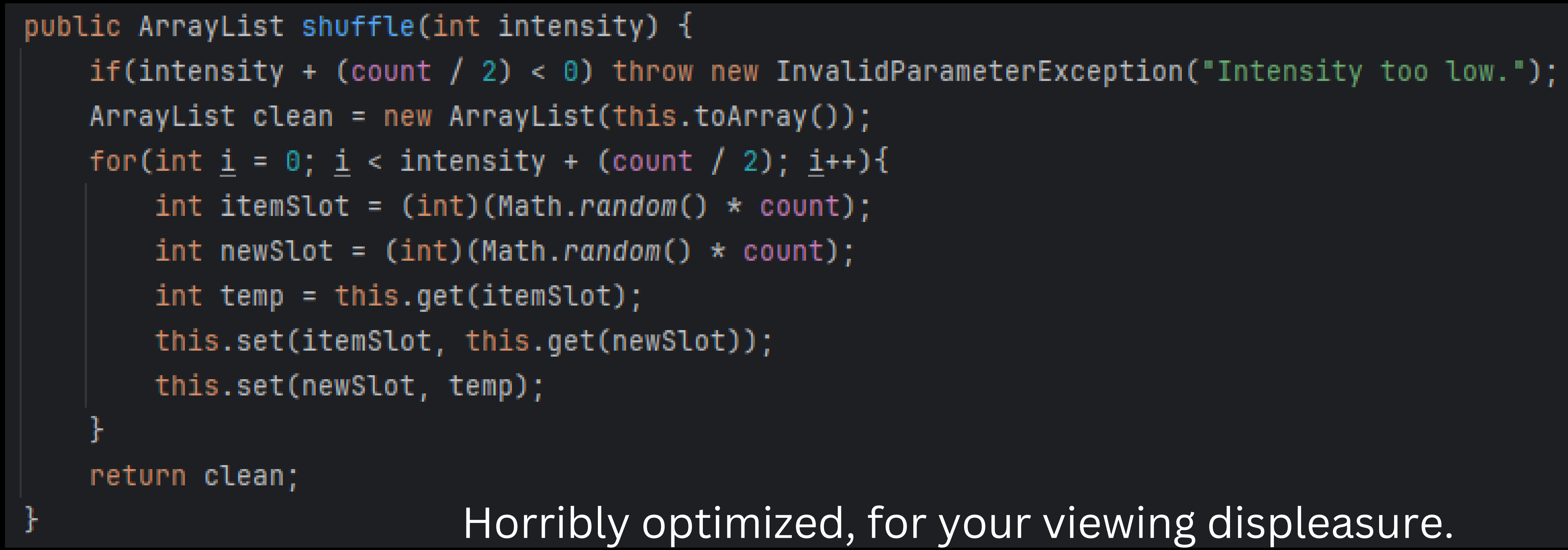
```java
@Override
public boolean contains(int value) {
    return indexOf(value) != -1;
}

/**
 * Gets the value at the specified index
 * @param index The index to look at.
 * @return The value at the index.
 * @throws ArrayIndexOutOfBoundsException Thrown when index is outside the range 0 - length.
 */
6 usages
@Override
public int get(int index) {
    if(index > this.count || index < 0) throw new ArrayIndexOutOfBoundsException();
    return list[index];
}

/**
 * Returns the index of the specified value.
 * @param value The value to find.
 * @return -1 if the value is not present, otherwise, the index of the value.
 */
4 usages
@Override
public int indexOf(int value) {
    for(int i = 0; i < count; i++){
        if(list[i] == value) return i;
    }
    return -1;
}
```

- Remove
- Set
- Size
- Sublist
- To Array
- Sort
- Is Sorted
- To String

Oh, yah, I added a Shuffle function too.

Mostly just cuz I could.

It takes an int “intensity” that determines how randomized it’s going to be.
If it’s less than -(size / 2), then it throws an error
It returns the unshuffled ArrayList, since it changes the ArrayList itself.



```
public ArrayList shuffle(int intensity) {
    if(intensity + (count / 2) < 0) throw new InvalidParameterException("Intensity too low.");
    ArrayList clean = new ArrayList(this.toArray());
    for(int i = 0; i < intensity + (count / 2); i++){
        int itemSlot = (int)(Math.random() * count);
        int newSlot = (int)(Math.random() * count);
        int temp = this.get(itemSlot);
        this.set(itemSlot, this.get(newSlot));
        this.set(newSlot, temp);
    }
    return clean;
}
```

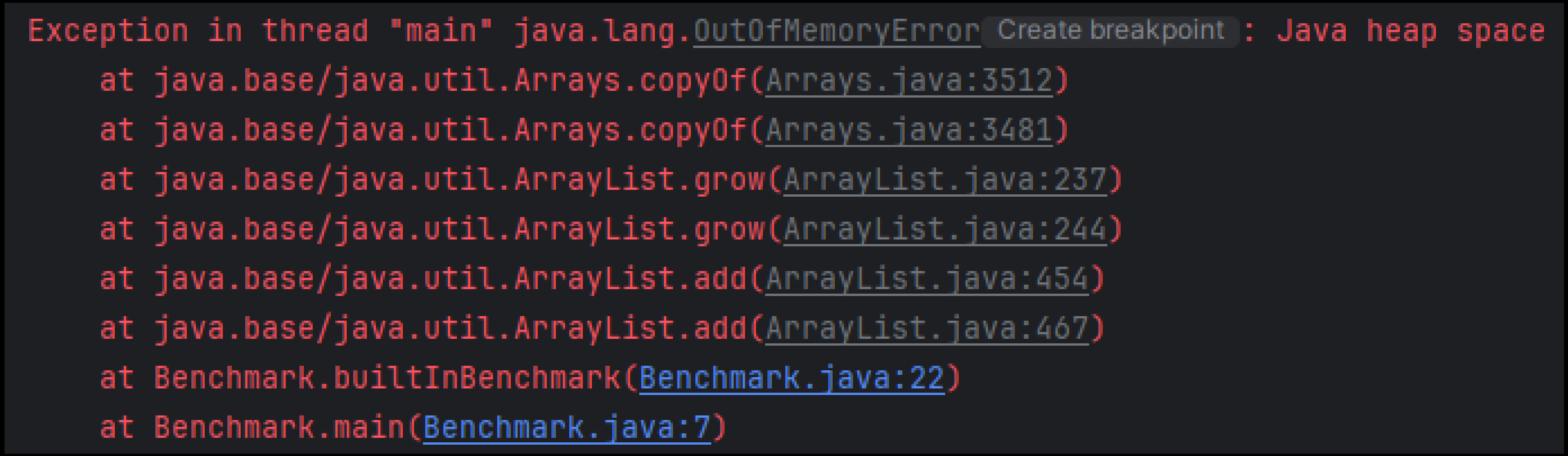
Horribly optimized, for your viewing displeasure.

Not quite sure what else you need.

Really shoulda been more specific when you said “PDF Presentaiton”

I, for one, think this is great.

In conclusion,



```
Exception in thread "main" java.lang.OutOfMemoryError : Java heap space
    at java.base/java.util.Arrays.copyOf(Arrays.java:3512)
    at java.base/java.util.Arrays.copyOf(Arrays.java:3481)
    at java.base/java.util.ArrayList.grow(ArrayList.java:237)
    at java.base/java.util.ArrayList.grow(ArrayList.java:244)
    at java.base/java.util.ArrayList.add(ArrayList.java:454)
    at java.base/java.util.ArrayList.add(ArrayList.java:467)
    at Benchmark.builtInBenchmark(Benchmark.java:22)
    at Benchmark.main(Benchmark.java:7)
```