I am started the test case early so that I can understand my code better inline with the challenge.

TEST CASE 1:

```
//TEST CASE 1 - No issues     PASS
entries.add("[1, 3]");
```

REACH HERE

1

3

1Added into innerlist: [[1, 3]]

1

3

**Final outer list: [[[1, 3]]]**

** Process exited - Return Code: 0 **

TEST CASE 2:

```
//TEST CASE 2 - No issues     PASS
entries.add("[1, 3]");
entries.add("[4, 7]");
```

REACH HERE

1

3

4

7

5Added into Inner List: [1, 3]

6Added into Inner List: [4, 7]

*****THE INNERLIST: [[1, 3], [4, 7]]

**Final outer list: [[[1, 3], [4, 7]]]**

** Process exited - Return Code: 0 **

TEST CASE 3:

```
//TEST CASE 3 - No issues     PASS
entries.add("[1, 3]");
entries.add("[2, 7]");
```

REACH HERE

1

3

2

7

4Added into Inner List: [1, 7]

4*****THE INNERLIST: [[1, 7]]

**Final outer list: [[[1, 7]]]**

** Process exited - Return Code: 0 **

TEST CASE 4:

```
//TEST CASE 4 - No issues     PASS
entries.add("[1, 3]");
entries.add("[4, 7]");
entries.add("[8, 9]");
```

REACH HERE

1

3

4

7

5Added into Inner List: [1, 3]

6Added into Inner List: [4, 7]

*****THE INNERLIST: [[1, 3], [4, 7]]

8

9

2Added into Inner List: [8, 9]

**Final outer list: [[[1, 3], [4, 7], [8, 9]]]**

** Process exited - Return Code: 0 **

## TEST CASE 5:

```
//TEST CASE 5 - No issues     PASS
entries.add("[1, 3]");
entries.add("[2, 7]");
entries.add("[8, 9]");
```

REACH HERE

1

3

2

7

4Added into Inner List: [1, 7]

4*****THE INNERLIST: [[1, 7]]

8

9

2Added into Inner List: [8, 9]

**Final outer list: [[[1, 7], [8, 9]]]**

** Process exited - Return Code: 0 **

## TEST CASE 6:  I have simply another object (4) and no issues

```
//TEST CASE 6 - No issues     PASS
entries.add("[1, 3]");
entries.add("[4, 7]");
entries.add("[8, 10]");
entries.add("[11, 25]");
```

REACH HERE

1

3

4

7

5Added into Inner List: [1, 3]

6Added into Inner List: [4, 7]

*****THE INNERLIST: [[1, 3], [4, 7]]

8

10

2Added into Inner List: [8, 10]

11

25

2Added into Inner List: [11, 25]

Final outer list: [[[1, 3], [4, 7], [8, 10], [11, 25]]]

** Process exited - Return Code: 0 **

TEST CASE 7:  I have simply another few objects (7) and no issues. This gives me confidence that there are no issues relating to non merging intervals

```
//TEST CASE 7 - No issues
entries.add("[1, 3]");
entries.add("[4, 7]");
entries.add("[8, 10]");
entries.add("[11, 25]");
entries.add("[27, 33]");
entries.add("[35, 38]");
entries.add("[42, 47]");
```

REACH HERE

1

3

4

7

5Added into Inner List: [1, 3]

6Added into Inner List: [4, 7]

*****THE INNERLIST: [[1, 3], [4, 7]]

8

10

2Added into Inner List: [8, 10]

11

25

2Added into Inner List: [11, 25]

27

33

2Added into Inner List: [27, 33]

35

38

2Added into Inner List: [35, 38]

42

47

2Added into Inner List: [42, 47]

Final outer list: [[[1, 3], [4, 7], [8, 10], [11, 25], [27, 33], [35, 38], [42, 47]]]

** Process exited - Return Code: 0 **

TEST CASE 8:  I have simply tried several merging intervals... I am starting easy at 3 FAIL

```
//TEST CASE 8 -
entries.add("[1, 3]");
entries.add("[2, 7]");
entries.add("[6, 10]");
//entries.add("[9, 25]");
//entries.add("[24, 33]");
//entries.add("[30, 38]");
//entries.add("[34, 44]");
```

REACH HERE

1

3

2

7

4Added into Inner List: [1, 7]

4*****THE INNERLIST: [[1, 7]]

6

10

3Added into Inner List: [[[1, 7]], 10]

Final outer list: [[[[[1, 7]], 10]]]

** Process exited - Return Code: 0 **

This is my first fail case.. And now I am bit more experienced to understand impact changes..
It was readily fixed, I had just had incomplete code...

TEST CASE 8v1:  PASS

3Added into Inner List: [1, 10]

Final outer list: [[[1, 10]]]

I am now going to try a long sequence of overlapping intervals...

TEST CASE 9:  PASS

```
//TEST CASE 9 -                PASS
entries.add("[1, 3]");
entries.add("[2, 7]");
entries.add("[6, 10]");
entries.add("[9, 25]");
entries.add("[24, 33]");
entries.add("[30, 38]");
entries.add("[34, 44]");
```

3Added into Inner List: [1, 44]

Final outer list: [[[1, 44]]]

TEST CASE 10:  This is now trying to mix overlapping and non-overlapping intervals

PASS

```
//TEST CASE 10
entries.add("[1, 3]");
entries.add("[4, 7]");
entries.add("[6, 9]");
entries.add("[8, 25]");
```

REACH HERE

THIS IS OBJECT: [1, 3] Counter: 1

1

3

THIS IS OBJECT: [4, 7] Counter: 2

4

7

5Added into Inner List: [1, 3]

6Added into Inner List: [4, 7]

*****THE INNERLIST: [[1, 3], [4, 7]]  // this is correct

THIS IS OBJECT: [6, 9] Counter: 3

END OF NEW INTERVAL: 7

6

9

START FIRST INTERVAL: 6

Last item inner list: [4, 7]

STARTINTERVAL: 4

Removed last item from innerList: [4, 7]  //this is correct

3Added into Inner List: [4, 9]  //this is correct

THIS IS OBJECT: [8, 25] Counter: 4  //this is correct

END OF NEW INTERVAL: 9

8

25

START FIRST INTERVAL: 8

Last item inner list: [4, 9]

STARTINTERVAL: 4

Removed last item from innerList: [4, 9]

3Added into Inner List: [4, 25]

Final outer list: [[[1, 3], [4, 25]]]  //this is correct

** Process exited - Return Code: 0 **

## TEST CASE 11:  A wider mix of intervals

```
//TEST CASE 11
entries.add("[1, 3]");
entries.add("[2, 7]");
entries.add("[8, 9]");
entries.add("[19, 45]");
entries.add("[40, 52]");
entries.add("[58, 61]");
```

2Added into Inner List: [58, 61]

Final outer list: [[[1, 7], [8, 9], [19, 52], [58, 61]]]

## TEST CASE 12:  Trying invalid test data.  The output is a fair reflection…

```
//TEST CASE 12 - Invalid data   (same interval first two objects)
entries.add("[1, 3]");
entries.add("[3, 9]");
entries.add("[8, 9]");
```

This is innerlist: [[1, 3], [3, 9]]

**Final outer list: [[[1, 3], [3, 9]]]**

## TEST CASE 13:  Trying invalid test data.  The output is a fair reflection…

```
//TEST CASE 13 - Invalid data
entries.add("[1, 3]");
entries.add("[3, 9]");
entries.add("[8, 12]");
entries.add("[12, 15]");
```

3Added into Inner List: [3, 15]

This is innerlist: [[1, 3], [3, 15]]

**Final outer list: [[[1, 3], [3, 15]]]**

TEST CASE 14:  Trying invalid test data.  Need to add validation code to ensure no end user incorrect inputs… But also, it is still performing output which is semantically inline with the design…

```
//TEST CASE 14 - Invalid data
entries.add("[1, 3]");
entries.add("[4, 9]");
entries.add("[8, 4]");
entries.add("[9, 2]");
```

2Added into Inner List: [9, 2]

**Final outer list: [[[1, 3], [4, 4], [9, 2]]]**