Sort by binary ones

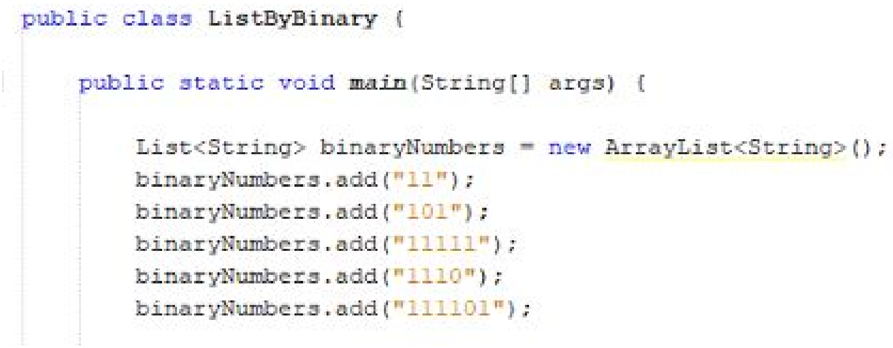
You are given a List of Strings that represents binary numbers.

Your task is to **sort the list based on the amount of 1's in the binary representation of each number.**

If two numbers have the same amount of 1's, the shorter string goes first. (ex: "11" goes before "101" when sorting 3 and 5 respectively)

Example

Create an ArrayList and populate it with Strings representing binary numbers:

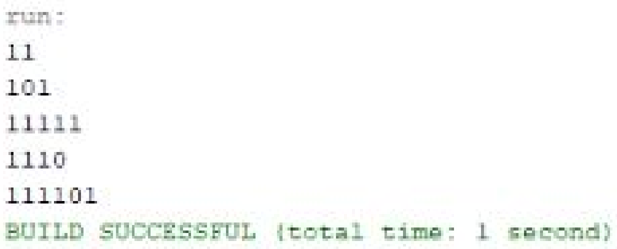


```
public class ListByBinary {

    public static void main(String[] args) {

        List<String> binaryNumbers = new ArrayList<String>();
        binaryNumbers.add("11");
        binaryNumbers.add("101");
        binaryNumbers.add("11111");
        binaryNumbers.add("1110");
        binaryNumbers.add("111101");
```

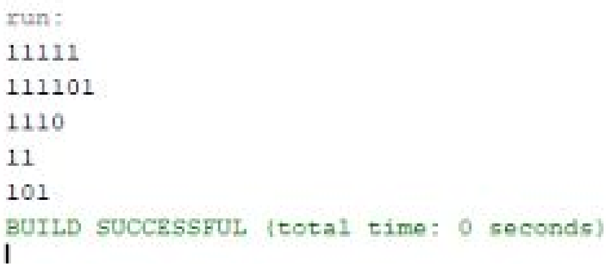
And print the content of the List. If the content is not sorted, the output will look like this:



```
run:
11
101
11111
1110
111101
BUILD SUCCESSFUL (total time: 1 second)
```

You will have to sort it according to the amount of 1's.

After sorting the output should look like this:



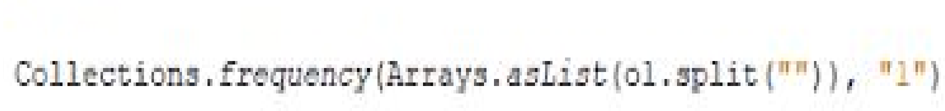
```
run:
11111
111101
1110
11
101
BUILD SUCCESSFUL (total time: 0 seconds)
```

The elements are sorted by number of 1's. The first element contains 5 1's and the last one 2. If the elements contains the same amount of 1's, the shortest String is first.

Tips

To sort the elements you need to use the Comparator class and override the Comparator sort method.

To calculate the amount of 1’s in the given String you can use the .frequency method.



```
Collections.frequency(Arrays.asList(o1.split("")), "1")
```

https://docs.oracle.com/javase/8/docs/api/java/util/Collections.html#frequency-java.util.Collection-java.lang.Object-

Have fun! :D