Given five positive integers, find the minimum and maximum values that can be calculated by summing exactly four of the five integers. Then print the respective minimum and maximum values as a single line of two space-separated long integers.

Example

$arr = [1, 3, 5, 7, 9]$

The minimum sum is $1 + 3 + 5 + 7 = 16$ and the maximum sum is $3 + 5 + 7 + 9 = 24$. The function prints

```
16 24
```

Function Description

Complete the miniMaxSum function in the editor below.

miniMaxSum has the following parameter(s):

- arr: an array of 5 integers

Print

Print two space-separated integers on one line: the minimum sum and the maximum sum of 4 of 5 elements.

Input Format

A single line of five space-separated integers.

Constraints

$1 \le arr[i] \le 10^9$

Change Theme   Language Java 15 ⌄

```java
    /*
     * Complete the 'miniMaxSum' function below.
     *
     * The function accepts INTEGER_ARRAY arr as parameter.
     */

public static void miniMaxSum(List<Integer> arr) {
    long min = 0;
    long max = 0;
    Collections.sort(arr);
    for (int i = 1; i < arr.size(); i++){
        if(i < 5){
            max += arr.get(i);
        }
    }
    for (int i = 0;i < arr.size() - 1;i++){
        min += arr.get(i);

    }
    System.out.print(min + " " + max);

    }

}
```

Line: 44 Col: 9

Line: 44 Col: 9

⬆ Upload Code as File   ☐ Test against custom input   **Run Code**   **Submit Code**

Congratulations

You solved this challenge. Would you like to challenge your friends? f 𝕏 in   **Next Challenge**

✓ **Test case 0**

✓ Test case 1 🔒

✓ Test case 2 🔒

✓ Test case 3 🔒

✓ Test case 4 🔒

✓ Test case 5 🔒

✓ Test case 6 🔒

Compiler Message

Success

Input (stdin)   Download

```
1   1 2 3 4 5
```

Expected Output   Download

```
1   10 14
```

Time and space complexity

The time complexity of this code is dominated by the sorting (I remembered this from Binary search) , the other for loops will not matter with their time complexity since sorting an array (which is necessary for binary search for example) automatically makes it O(n*log n). The for loops iterate 1-4 (constants) since array size is less then 5 however they are dominated by the sorting method.