

Loyola University Chicago
Department of Computer Science
COMP 272: Data Structures II (Spring 2024)

Assignment # 6

This is an individual assignment.

Deadline: Monday, **April 8th, 2024, 11:55PM.**

You need to submit your solutions in Repl.it.

Sorting 1 (27 points)

You need to implement a method named *showDuplicates()* in the Main class that accepts an `ArrayList<String>` as an argument and returns an `ArrayList<String>`. The returned list should contain only the unique strings that appear more than once in the input list.

See comments in the code for the expected output format.

Sorting 2 (27 points)

You need to implement a method *pair()* to find pairs of numbers in an array that add up to a given sum `k`. It should take in an input array and number `k` then return a list of pairs that add up to the `k`-value.

See comments in the code for the expected output format

Priority Queue - Game (27 points)

You are given an array of integers of boulders where `boulders[i]` is the weight of the i^{th} boulder.

We are playing a game with the boulders. On each turn, we choose the heaviest two boulders and smash them together. Suppose the heaviest two boulders have weights `x` and `y`. The result of this smash is:

If `x == y`, both boulders are destroyed, and

If `x != y`, the boulder of weight `x` is destroyed, and the boulder of weight `y` has new weight `y - x`.

At the end of the game, there is at most one boulder left.

Return the weight of the last remaining boulder. If there are no boulders left, return 0.

Example 1:

Input: `stones = [2,7,4,1,8,1]`

Output: 1

Explanation:

We combine 7 and 8 to get 1 so the array converts to `[2,4,1,1,1]` then,

we combine 2 and 4 to get 2 so the array converts to `[2,1,1,1]` then,

we combine 2 and 1 to get 1 so the array converts to [1,1,1] then,
we combine 1 and 1 to get 0 so the array converts to [1] then that's the value of the last stone.

Example 2:

Input: stones = [1]

Output: 1

Priority Queue – Min Heap (19 points)

You need to implement two methods: *add()* and *contains()*. Please consult the documentation contained within the code block for further instruction on best practices to complete this task.

Recommend reading the entire class's code to understand how the class implements the minheap. If understanding how the structure works and the methods there, very little code should be written.

Submission:

- 1) Before submission, make sure your code passes all the JUnit tests. Keep in mind, however, that passing the test cases does not guarantee that your code is correct or efficient. Your assignment will be graded considering test results, correctness, and efficiency.
- 2) The submission should be completed in Replit.
- 3) Include your name and class number as comments at the top of each submitted Java file.