

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

Assignment # 7

This is an individual assignment.

Deadline: Tuesday, **April 16th, 2024, 11:55PM.**

You need to submit your solutions in Repl.it.

Sorting Exercise 1 (25 points)

Complete the implementation of the LUCSorter class to sort an array of integers in descending order using the selection sort algorithm. Specifically, you will need to fill in the sort method to iterate over the array and select the maximum element to move to the front, and the **maximumPosition** method to find the index of the maximum element from a given starting position. Ensure your LUCSorter class with the completed sort and **maximumPosition** methods passes all tests.

Sorting Exercise 2 (25 points)

Modify the LUCSorter class to complete the merge method that combines two sorted subarrays, ensuring that all elements that are divisible by a given integer k are at the front of the merged array. Ensure your LUCSorter class with the completed merge method passes all tests.

Destroying Asteroids 4 (25 points)

You are given an integer '**mass**', which represents the original mass of a planet. You are further given an integer array '**asteroids**', where **asteroids[i]** is the mass of the i^{th} asteroid.

You can arrange for the planet to collide with the asteroids in any arbitrary order. If the mass of the planet is greater than or equal to the mass of the asteroid, the asteroid is destroyed and the planet gains the mass of the asteroid. Otherwise, the planet is destroyed.

Return true if all asteroids can be destroyed. Otherwise, return false.

Example 1:

Input: mass = 10, asteroids = [3,9,19,5,21]

Output: true

Explanation: One way to order the asteroids is [9,19,5,3,21]:

- The planet collides with the asteroid with a mass of 9. New planet mass: $10 + 9 = 19$
- The planet collides with the asteroid with a mass of 19. New planet mass: $19 + 19 = 38$
- The planet collides with the asteroid with a mass of 5. New planet mass: $38 + 5 = 43$
- The planet collides with the asteroid with a mass of 3. New planet mass: $43 + 3 = 46$
- The planet collides with the asteroid with a mass of 21. New planet mass: $46 + 21 = 67$

All asteroids are destroyed.

Example 2:

Input: mass = 5, asteroids = [4,9,23,4]

Output: false

Explanation:

The planet cannot ever gain enough mass to destroy the asteroid with a mass of 23.

After the planet destroys the other asteroids, it will have a mass of $5 + 4 + 9 + 4 = 22$.

This is less than 23, so a collision would not destroy the last asteroid.

Constraints:

$1 \leq \text{mass} \leq 10^5$

$1 \leq \text{asteroids.length} \leq 10^5$

$1 \leq \text{asteroids}[i] \leq 10^5$

Number of Rescue Boats (25 points)

You are given an array **people** where **people[i]** is the weight of the **ith** person, and an infinite number of canoes where each canoe can carry a maximum weight of limit. Each canoe carries at most two people at the same time, provided the sum of the weight of those people is at most limit. Return the minimum number of canoes to carry every given person.

Example 1:

Input: people = [1,2], limit = 3

Output: 1

Explanation: 1 boat (1, 2)

Example 2:

Input: people = [3,2,2,1], limit = 3

Output: 3

Explanation: 3 boats (1, 2), (2) and (3)

Example 3:

Input: people = [3,5,3,4], limit = 5

Output: 4

Explanation: 4 boats (3), (3), (4), (5)

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.