

American University of Armenia, CSE
CS121 Data Structures B
Spring 2022

Homework Assignment 2

Due Date: Saturday, February 19 by 23:59 electronically on Moodle

Please solve the programming tasks using Java, following good coding practices (details are on Moodle). Don't forget to submit the corresponding assignment report.

1. **(15 points)** It's 18:00, year 202x, Arcane season 2 is out, and you are going to visit a friend to binge-watch it together. You want to be at the friend's house at 22:00 the latest. However, you still haven't submitted your data structures homework. You decide that you will try to solve as many tasks as possible before leaving. It takes you t minutes to reach the friend's house and the j -th task (assuming an order from 1 to n) takes $j * 5$ minutes to solve.

Write an efficient non-recursive method that will calculate the number of problems that you can solve before leaving. The inputs to the method are the number of problems denoted by n , and number of minutes t you need to reach the destination.

2. **(15 points)** You have a box with n columns ($1 \leq n \leq 100$) each containing k_i ($1 \leq k_i \leq 100$) metallic cubes put on top of each other. If you hold a magnet next to a side of the box, the cubes will be pulled towards that side as shown in Figure 1. You are given input values n —the number of columns of the box, and n numbers k_i representing the initial number of cubes in each column respectively. Write an efficient program that outputs the number of cubes in each column after holding a magnet at the left side of the box. What is the runtime of this method?

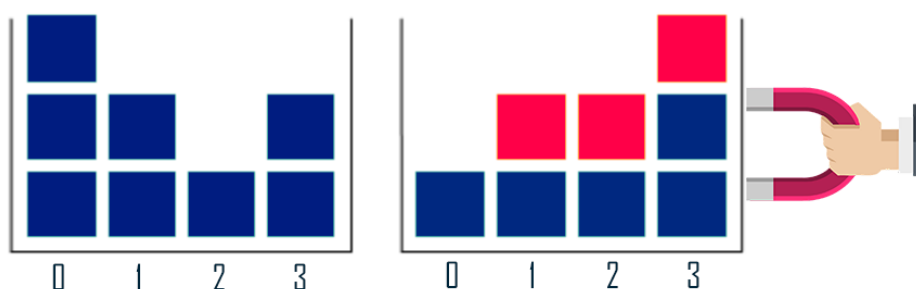


Figure 1: Left: initial Box. Right: the same box after putting a magnet from the right

3. (15 points) Consider a sequence of lowercase Latin letter Strings of maximum 7 characters. Implement the *radix sort* algorithm for such a sequence to sort them lexicographically. You can use an array to represent the sequence to be sorted, but should use singly linked lists for the intermediate steps of the algorithm, i.e. the buckets should be singly linked lists.
4. (15 points) Didn't I warn you not to get involved with the Russian mafia? Now, look at you. Sitting here, in front of Ivan, who suggests that you play a traditional fun game—**Russian Roulette**. You sure know the rules of this deadly(?) game; one or more (up to six) paint bullets are loaded into a revolver at random positions, and the cylinder is randomly rotated. After that, both sides take turns pointing the gun at themselves and pulling the trigger. The last man standing clean is the winner. Write a method that takes as an argument the number of bullets to be inserted into the revolver and the names of the two players and simulates the game. Your method should set up the revolver using an appropriate data structure, should print the positions of bullets right after rotation, and should print out the information about the player status after each shot.
5. (35 points) The task is to implement various sorting algorithms for a doubly linked list of integers. For the sake of this exercise, make the Node class package-private, and make *header* protected.
- (a) Implement a generic method that given the header node of a doubly linked list *L* prints its contents.
 - (b) Implement a method *selectionSort* that given the header node of an Integer doubly linked list *L*, sorts it using *selection sort*. You are **not allowed** to create any arrays, nodes or list objects. You are not allowed to swap the values (elements) within the nodes. At each iteration, you should insert the node with the smallest value at the current position (i.e., you should not swap nodes).
 - (c) Implement a method *insertionSort* that given the header node of an Integer doubly linked list *L*, sorts it using *insertion sort*. You are **not allowed** to create any arrays, nodes or list objects. You are not allowed to swap the values (elements) within the nodes. At each iteration, you should insert the current node into its correct position within the sorted portion (i.e., you should not swap nodes).
 - (d) Implement a method *bubbleSort* that given the header node of an Integer doubly linked list *L*, sorts it using *bubble sort*. You are **not allowed** to create any arrays, nodes or list objects. You are not allowed to swap the values (elements) within the nodes.