CS211FZ (Algorithms & Data Structures 2)

Assignment 4

Chris Roadknight

chris.roadknight@mu.ie

Released date: Sunday, 25th May 2025

Deadline: Friday, 30th May 2025, at 18:00 Beijing time

This is an open-book, graded assignment. You may only use the module's slides and textbooks to help with the Assignment. Please cite all references as comments in your submissions. You cannot directly reuse a solution from online sources or AI. You must not engage with another student, in person or electronically (via phone, social media, etc.), to secure assistance with this Assignment. If you do so (even for only one of the questions), you will receive an automatic failure (0%), and it will also be reported to the Executive Vice-Dean of MIEC and/or Maynooth University Plagiarism board. We will perform similarity checks on submitted assignments to check for collaborative efforts. The lecturer reserves the right to interview you about your submission in special cases.

Question 4.1

John wants to have a journey around Ireland. He has many items that he likes to carry with him, but although he has three suitcases, unfortunately, he cannot accommodate all his items in the suitcases. Thus, he needs to decide which items to pick. He knows the size and worth of each item. Therefore, he likes to pick the items so that the total values of the selected items be maximum. There are several items that he cannot take part of them, like his violin. But some of them, he can take part of them like sugar.

You need to write a program in Java to help John select his items and tell John that each item should be put in which suitcase.

The input of the program is the number of suitcases that John has and also the capacity of each of them. In addition, the total number of the items, the size and worth of each, and whether John can take a part of the item.

Sample inputs:

The number of suitcases: 3

The capacity of suitcase 1 in cm^3: 90

The capacity of suitcase 2 in cm^3: 150

The capacity of suitcase 3 in cm^3: 100

The total number of items: 10

The volume of item 1 in cm^3: 40

The worth of item 1 in Euro: 10

Can take some part of item 1: Yes

The volume of item 2 in cm^3: 50

The worth of item 2 in Euro: 5

Can take some part of item 2: Yes

The volume of item 3 in cm^3: 70

The worth of item 3 in Euro: 5

Can take some part of item 3: No

The volume of item 4 in cm^3: 40

The worth of item 4 in Euro: 5

Can take some part of item 4: Yes

The volume of item 5 in cm^3: 80

The worth of item 5 in Euro: 2

Can take some part of item 5: No

The volume of item 6 in cm^3: 30

The worth of item 6 in Euro: 5

Can take some part of item 6: Yes

The volume of item 7 in cm^3: 35

The worth of item 7 in Euro: 20

Can take some part of item 7: Yes

The volume of item 8 in cm^3: 40

The worth of item 8 in Euro: 5

Can take some part of item 8: No

The volume of item 9 in cm^3: 73.5

The worth of item 9 in Euro: 5

Can take some part of item 9: No

The volume of item 10 in cm^3: 150

The worth of item 10 in Euro: 5

Can take some part of item 10: No

Output:

The output should tell the user which items should be put to each of the suitcases so that the worth of the selected items be maximised, it should also show the total value of items in the suitcases

Submission Notes:

- A template SuitcasePack.java is available on moodle, this template is only one possible approach and you are not required to use it
- You need to create an efficient algorithm and then analyse it. Thus, you need to consider both the correct algorithm and also an efficient approach. In your analyse, you need to mention which algorithm design technique (divide-and-conquer, greedy, etc.) you used and why. In addition, you should calculate the running time of your algorithm.
- You need to explain your algorithm in detail.
- You need to provide at least one sample test case that assigns values to variables automatically in your code that do not need user input. We need this feature to help you and TAs evaluate your code faster and more easily. Thus, getting input from the user is not necessary.
- Your code should be fully commented.
- Your final submission should be one pdf file that includes your software AND a one page description of your approach covering all of the above

Important submission details

Please indicate the Operating System (Linux/Windows/MacOS/Online), IDE (e.g. Eclipse, Visual Studio Code), and Java SDK version used for testing in your submission. If you use an online IDE, please specify the IDE (http://repl.it) and provide a link where possible.

Software MUST be shown to your TA before the end of the session. All work must be submitted via Moodle (see "Assignments" section for submission). Work submitted via other means will not be

accepted unless you have prior arrangements with the lecturer. All work MUST be submitted by the due date deadline. Late submissions will not be accepted.

Your submission should be one single PDF file, also including your Java codes. Submitting in any other format cannot be accepted.

Note: You should type your answers in a text editor like Microsoft Word, and then convert it to PDF. You MUST NOT take pictures from answers and then make a PDF from the pictures.