2021-09-16 - Pearls of Computer Science Core - Algorithmics Diagnostic

Course: B-CS-MOD01-1A-202001021/202001022 B-CS Pearls of Computer Science Module - 202001021/202001022

Duration:	1 hour
Generated on:	Sep 16, 2021

Contents:

Pages:

2021-09-16 - Pearls of Computer Science Core - Algorithmics Diagnostic

Course: B-CS Pearls of Computer Science Module - 202001021/202001022

Welcome to the digital diagnostic test for Pearl 001 Algorithmics.

- You may use **1 A4 sheet with your own notes** for this test, as well as a simple calculator
- **Scientific or graphical calculators, laptops, mobile phones, books etc. are not allowed. Put those in your bag now (with the sound switched off).**
- Please use the **BBB/Canvas chat for any question** during the exam.
- You have enough time to familiarize yourself with the Remindo environment. Make good use of it! The real test has more questions (100 points total).

Total number of points: 40

Number of questions: 5

1 Suppose you execute the following assignments in Python

5 pt.

```
room = ["Hopper", "Turing", "Lovelace"]
capacity = [129, 112, 236]
```

Here *room* is a list of room names, and *capacity* is a list of integers.

Write a Python condition (*not* an **if** statement) that tests whether room *i* has the largest capacity of the three rooms (note that you don't need to know *i*).

2 Suppose you execute the following assignments in Python

5 pt.

```
room = ["Hopper", "Turing", "Lovelace"]
capacity = [129, 112, 236]
```

Here *room* is a list of room names, and *capacity* is a list of integers.

Assign to a new list *largest* the name and the capacity of the room with the highest capacity.

3 Suppose you execute the following assignments in Python

5 pt.

```
room = ["Hopper", "Turing", "Lovelace"]
capacity = [129, 112, 236]
```

Here *room* is a list of room names, and *capacity* is a list of integers.

Write a sequence of assignments that is as short as possible, resulting in a change to *capacity* after which the integers are ordered from lowest to highest. (It is *not* correct to assign an entirely new value to *capacity*; You must modify the list by swapping elements.)

4 Consider the following list

10 pt.

```
[13, 1, 18, 3, 21, 19]
```

Show how merge sort sorts this list, by presenting how the list is split and zipped back together, i.e. write down every change the algorithm makes to the list in a new line.

5 Assume there is a global pandemic, and to protect yourself and others from an infection, you have gathered a large collection of face masks in your wardrobe. The masks are ordered from small to big. One day you're feeling brave, and want to order them from big to small instead!

10 pt. **a.** Provide an algorithm in *human language* with **unambiguous** and **numbered** instructions that yields the desired outcome as fast as possible. Your instructions may refer to at most 2 masks at the same time -- Never an arbitrary number of them.

You may assume that you have ample space outside your wardrobe to re-arrange the masks in any way you like, but the algorithm needs to stop with every mask being at the correct place inside the wardrobe.

Do not give an answer in Python!

5 pt. **b.** How many steps does your algorithm take in terms of number of masks n .

Thank you! Your test has been saved. You can check Canvas for the solution of the questions.

Please remember that the **real test tomorrow has more questions** (100 points total rather than the 40 points you saw here)!