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 6, 2021

Contents: Pages:

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

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).
- For *technical* questions (concerning the chromebooks, Remindo etc.): raise your mouse
- For *pearl content* question: raise your hand
- 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

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)!

Correction model

1.
5 pt.

Correction criterion	Points
-2 if an "if" is included -1 for missing 'and' -3 for the wrong list -3 for using > rather than >= -2 for other syntax mistakes Answer: capacity[i] >= capacity[0] and capacity[i] >= capacity[1] and capacity[i] >= capacity[2] In particular, avoid solving this for general lists/list sizes. The question very specifically asks to provide a condition that tests whether "room i" hast the largest capacity of the *three* rooms. No need to think of a solution for list size >3.	5 points
Total points:	*5 points*

2.
5 pt.

Correction criterion	Points
This often yields 0 points, if it is not quite correct. -4 if 'append' is used -2 for wrong brackets -3 for missing brackets -1 for one missing bracket -2 for a 2-dimensional list -3 for absence of any index Answer: largest = [room[2], capacity[2]]	5 points
Total points:	*5 points*

3.
5 pt.

Correction criterion	Points
-2 for every additional assignment beyond the first 0pts if capacity is assigned a new list -2 for wrong order -3 for wrong list (room names) Answer: capacity[0], capacity[1] = capacity[1], capacity[0]	5 points
Total points:	*5 points*

4.
10 pt.

Correction criterion	Points
This question is strictly assessed. -3 if pairs are not split -5 if zipping is not included -5 if lists are mixed up up to -8 for systematic mistakes Answer: [13, 1, 18, 3, 21, 19] is split into [13, 1, 18] and [3, 21, 19] - [13, 1, 18] is split into [13] and [1, 18] -- [1, 18] is split into [1] and [18] -- [1] and [18] are merged into [1, 18] - [13] and [1, 18] are merged into [1, 13, 18] - [3, 21, 19] are split into [3] and [21, 19] -- [21, 19] are split into [21] and [19] -- [21] and [19] are merged into [19, 21] - [3] and [19, 21] are merged into [3, 19, 21] [1, 13, 18] and [3, 19, 21] are merged into [1, 3, 13, 18, 19, 21]	10 points
Total points:	*10 points*

5.
15 pt.

a.

Correction criterion	Points
-2 to -6 for ambiguities -5 if an answer is given in Python -8 if more than two masks are used in an instruction, e.g. 'Take all masks and put them in reversed order' Answer: An example algorithm could look like the one below, but every answer that adheres to the restrictions and fulfills the desired outcome gets awarded full points: We assume there is a table to temporarily store the masks outside: 1. Take the right-most mask out of the wardrobe and put it right of the most recently placed mask on the table, or to the very left if the table is still empty 2. If there are no more masks in the wardrobe, go to step 3; else go to step 1 3. Take the left-most mask from the table and place it to the right of the most recently placed mask in the wardrobe, or to the left-most position of the wardrobe is empty 4. If there are no more masks on the table we are done, else we go to step 3	10 points
Total points:	*10 points*

b.

Correction criterion	Points
This gives 0 or 5 points, hardly anything in between. Points are given if (b) reflects the steps of the algorithm in (a) correctly, even if (a) is wrong. Answer: For the example algorithm above, it takes 2n steps, given that each of the n masks is picked up twice.	5 points
Total points:	*5 points*

Caesura

Points scored	Grade
40	10
39	9.8
38	9.5
37	9.3
36	9.0
35	8.8
34	8.5
33	8.3
32	8.0
31	7.8
30	7.5
29	7.3
28	7.0
27	6.8
26	6.5
25	6.3
24	6.0
23	5.8
22	5.5
21	5.3
20	5.1
19	4.9
18	4.7
17	4.5
16	4.3
15	4.1
14	3.9
13	3.7
12	3.5
11	3.3
10	3.0
9	2.8
8	2.6
7	2.4
6	2.2
5	2.0
4	1.8
3	1.6
2	1.4
1	1.2
0	1.0

Question identifiers

These identifiers can be used to track the exact origin of the question. Use these identifiers together with the identifier of this document when sending in comments about the questions, so that your comment can be connected precisely with the question you are referring to.

Document identifier: 6449-7999

Question number	Question identifier	Version identifier
1	34017	b57963de-dd7c-3e74-faa2-cea6bd2e6448
2	34022	5def5464-8581-2224-48cc-53676f21834a
3	34027	e026f723-96e2-8e2a-465f-6ffe9a249a94
4	34047	20d774bf-fb63-2961-f58a-1adef78d0987
5	34032	20548e39-cca3-2ace-acd3-061f15d1ebc4