

2020-09-11 - Pearls of Computer Science Core - Pearl 001

Course: B-CS-MOD01-1A-202001022 B-CS Pearls of Computer Science Core
202001022

Duration: 1 hour
Number of questions: 10
Generated on: Sep 8, 2020

Contents:	Pages:
▪ A. Front page	1
▪ B. Questions.....	5
▪ C. Answer form	7
▪ D. Correction model	8

2020-09-11 - Pearls of Computer Science Core - Pearl 001

Course: B-CS Pearls of Computer Science Core 202001022

Welcome to the digital exam 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
- Total number of points: 100

Number of questions: 10

You can score a total of 100 points for this exam, you need 55 points to pass the exam.

1 Question 1a

5 pt.

Suppose you execute the following assignments in Python

```
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 Question 1b

5 pt.

Suppose you execute the following assignments in Python

```
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 Question 1c

5 pt.

Suppose you execute the following assignments in Python

```
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 Question 2

Consider the following Python function:

```
01. def compute(data):
02.     even = []
03.     odd = []
04.     i = 0
05.     while i < len(data):
06.         if data[i] % 2 == 0:
07.             even.append(data[i])
08.         else:
09.             odd.append(data[i])
10.         i = i + 1
11.     return [odd, even]
```

- 5 pt. a. What is the return value upon calling compute([8, 21, 9, 19, 13, 1, 14])?
- 5 pt. b. Using your own words, what does the algorithm do?

5 Question 3

10 pt.

Consider again the Python function given in Question 2:

```
01. def compute(data):
02.     even = []
03.     odd = []
04.     i = 0
05.     while i < len(data):
06.         if data[i] % 2 == 0:
07.             even.append(data[i])
08.         else:
09.             odd.append(data[i])
10.         i = i + 1
11.     return [odd, even]
```

Assume we input a list *data* of length n . How many steps does compute need to finish?

1. Approximately n
2. Approximately n^2
3. Approximately $n \cdot \log_2 n$
4. Approximately \sqrt{n}

Motivate your answer.

6 Question 4

Consider again the Python function given in Question 2:

```
01. def compute(data):
02.     even = []
03.     odd = []
04.     i = 0
05.     while i < len(data):
06.         if data[i] % 2 == 0:
07.             even.append(data[i])
08.         else:
09.             odd.append(data[i])
10.         i = i + 1
11.     return [odd, even]
```

Provide short explanations to the questions below and motivate your answers.

- 5 pt. a. What happens if line 10 is deleted?
- 5 pt. b. What happens if the condition $i < len(data)$ in line 5 is replaced with $i \leq len(data)$?
- 5 pt. c. What happens if $i = 0$ in line 4 is replaced with $i = 1$?

7 Question 5a

10 pt.

Consider the following list

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

Show how bubble sort sorts this list, by writing down the list after every single modification.

8 Question 5b

10 pt.

Consider the following list

```
[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.

9 **Question 6**

15 pt.

You are given two (unordered) lists with student names belonging to two student houses. You want to find out whether the lists are the same. Consider two ways of doing this:

1. Sort each list by name and compare both lists element by element, and
2. For each name on list 1 try to find the matching name on list 2.

Assuming the lists are very long, which of the two methods is faster? Explain your answer as accurately as possible.

10 **Question 7**

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, 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 amount 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 amount of masks n .

Thank you, your exam has been saved. You will be notified about your grade after your answers were checked by the correction team. Stay safe!