2022-09-15 - Pearls of Computer Science Core - Algorithmics Diagnostic

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

Contents: Pages:

Duration:	1 hour
Generated on:	Sep 15, 2022

2022-09-15 - Pearls of Computer Science Core - Algorithmics Diagnostic

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

Welcome to the ungraded 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: 35

1
5 pt.

Question 1a

Suppose you execute the following assignments in Python

```
clubs = ["Dance", "Theatre", "Sports"]
members = [68, 162, 92]
```

Here *clubs* is a list of club names at the University of Twente, and *members* is a list of the corresponding (fictional) number of active members in each club.

Write a Python condition (*not* an **if** statement) that tests whether the "Dance" club is the smallest of the three clubs.

2
5 pt.

Question 1b

Suppose you execute the following assignments in Python

```
clubs = ["Dance", "Theatre", "Sports"]
members = [68, 162, 92]
```

Here *clubs* is a list of club names at the University of Twente, and *members* is a list of the corresponding (fictional) number of active members in each club.

Assign to a new list '*smallest*' both the name and the number of active members of the club with the fewest active members.

3
5 pt.

Question 1c

Suppose you execute the following assignments in Python

```
clubs = ["Dance", "Theatre", "Sports"]
members = [68, 162, 92]
```

Here *clubs* is a list of club names at the University of Twente, and *members* is a list of the corresponding (fictional) number of active members in each club.

Write a sequence of assignments that is **as short as possible**, resulting in a change to '*members*' after which the number of active members are ordered from lowest to highest.

(NB: It is *not* correct to assign an entirely new value to *members*. You *must* modify the list by swapping elements.)

4
10 pt.

Question 5a

Consider the following list

```
[8, 20, -4, 4, -18, 6]
```

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

5
10 pt.

Question 6

After having studied all week for the Algorithmics test you lie awake one night and wonder if one could get an even faster search algorithm than binary search.

Then it dawns on you: *ternary search!* Rather than dividing a list into two equal parts, *ternary search* divides the list into *three* equal parts (left, middle, right) and then determines in which of these three parts the search value lies (if any).

Can this idea be realized, ie. could such an algorithm work in practice?

If not, why not?

If so, what would be the complexity of *ternary* search -- In particular, would it be fundamentally faster than binary search? Explain your answer.

Thank you! Your answers were 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 35 points you saw here)!

Correction model

1.
5 pt.

Correction criterion	**Points**
-2 if an "if" is included -2 for missing 'and' -3 for the wrong list (clubs vs members) -3 if the largest club was used instead of the smallest (< vs >) -1 if the same element was used for comparison and everything else is correct, eg. members[0] < members[1] and members[0] < members[1] -2 for syntax mistakes, eg. =< instead of <= Answer: members[0] < members[1] and members[0] < members[2]	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' or 'extend' are used -3 for wrong or missing brackets -1 for one missing bracket -3 for absence of any index Answer: smallest = [clubs[0], members[0]] NB: Forming an entirely new list is only partially correct. The intent is to access the existing lists.	5 points
Total points:	*5 points*

3.
5 pt.

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

4.
10 pt.

Correction criterion	**Points**
The question is assessed strictly: -3 for every mistake, eg. skipping a step up to -8 for a systematic error [8, 20, -4, 4, -18, 6] [8, -4, 20, 4, -18, 6] [8, -4, 4, 20, -18, 6] [8, -4, 4, -18, 20, 6] [8, -4, 4, -18, 6, 20] [-4, 8, 4, -18, 6, 20] [-4, 4, 8, -18, 6, 20] [-4, 4, -18, 8, 6, 20] [-4, 4, -18, 6, 8, 20] [-4, -18, 4, 6, 8, 20] [-18, -4, 4, 6, 8, 20]	10 points
Total points:	*10 points*

5.
10 pt.

Correction criterion	**Points**
Points are given for the explanation rather than the precise answer. Points are awarded leniently as long as the argument makes sense (ie. no precise calculation like below is necessary). A sensible explanation may yield up to 8 points even if the answer is not correct. Conversely, just writing "yes" or "no" does not yield more than 2 points, even if it is correct. Answer: Yes, this algorithm can be made to work. It's complexity would be c_3*log_3(n) in the length n of the list, compared to c_2*log_2(n) for binary search. Here c_2 and c_3 are constants. Even though the algorithm exists, it is not fundamentally faster than binary search. Going from binary to ternary the speedup is (c_2 * log_2(n))/ (c_3 * log_3(n)) = (c_2/c_3) log_2(3) = C Which is a constant (as the ratio of two logarithms with different bases is constant). Depending on the exact constants it is even very likely to be smaller than 1, implying that ternary is actually a slowdow. Determining if the searched-for value lies in left, middle or right requires two comparisons in the worst case. With the same number of comparisons binary search always divides the list into four compartments.	10 points
Total points:	*10 points*

Caesura

Points scored	Grade
35	10
34	9.7
33	9.4
32	9.1
31	8.9
30	8.6
29	8.3
28	8.0
27	7.7
26	7.4
25	7.1
24	6.9
23	6.6
22	6.3
21	6.0
20	5.7
19	5.4
18	5.2
17	5.0
16	4.7
15	4.5
14	4.3
13	4.0
12	3.8
11	3.6
10	3.3
9	3.1
8	2.9
7	2.6
6	2.4
5	2.2
4	1.9
3	1.7
2	1.5
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: 10498-13538

Question number	Question identifier	Version identifier
1	69019	b893a1f8-49f3-e81d-4b3a-9b6edbb162c9
2	69024	ee7c8c0f-fb8c-43bf-37e8-b0d3384bfdec
3	69029	2426ba42-402f-52b9-b73d-365ff8870e03
4	69034	a1e505a9-5082-f1a2-cc5a-cd33e85e4504
5	69044	47b82e00-54b7-8178-521e-87df62e1ff64