Review Test Submission: Midterm Test

User	MR JOHN WESLEY KOMMALA
Module	Algorithms and Data Structures I
Test	Midterm Test
Started	17/11/20 10:10
Submitted	17/11/20 11:00
Status	Completed
Attempt Score	92 out of 100 points
Time Elapsed	49 minutes out of 50 minutes

Instructions This test is **timed:** you have 50 min to submit it (1hr for students who have sent the LENS report to the lecturer).

You **must complete the test in one sitting:** once you start the test you cannot restart it.

Questions in this test are drawn from material covered during weeks 1-6 (prior to Study Week).

When you start the test you will encounter:

- A question asking you to confirm you understand College policies on academic honesty and plagiarism.
    - Please read plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar
        - Please complete the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write
- A question giving you a problem together with a program solving it. You will need to analyze the worst-case asymptotic running time performance of the program.
- A question asking you to compare different performance analyses and notations.

- A question asking you to choose between different Abstract Data Types in the solution of a problem.

Results Displayed	All Answers, Submitted Answers, Correct Answers, Feedback, Incorrectly Answered Questions

Question 1

0 out of 0 points

I understand that this is an individual assessment and that collaboration is not permitted. I have not received any assistance with my work for this assessment. Where I have used the published work of others, I have indicated this with appropriate citation.

I have not and will not share any part of my work on this assessment, directly or indirectly, with any other student.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at
http://www.tcd.ie/calendar

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at
http://tcd-ie.libguides.com/plagiarism/ready-steady-write

Selected Answer: True

Answers: True
False

Question 2

42 out of 50 points

Problem: Given an array *nums* of n integers and an integer *target*, find three integers in nums such that the sum is closest to target. Return the sum of the three integers. You may assume that each input would have exactly one solution.

Program:

```
public int threeSumClosest(int[] nums, int target) {
    int diff = Integer.MAX_VALUE, sz = nums.length;
    Arrays.sort(nums); // Here assume Heapsort
    for (int i = 0; i < sz && diff != 0; ++i) {
      int lo = i + 1, hi = sz - 1;
      while (lo < hi) {
        int sum = nums[i] + nums[lo] + nums[hi];
```

```
            if (Math.abs(target - sum) < Math.abs(diff))
              diff = target - sum;
            if (sum < target)
              ++lo;
            else
              --hi;
          }
        }
        return target - diff;
    }
```

- Give a **tight upper bound** for the worst-case asymptotic running time of the code as a function of the input size. *[15% of the mark for this question]*
- Explain your answer. In your explanation you should include
  - a description of a **worst-case input of size N**; *[20% of the mark for this question]*
  - a **detailed analysis** that justifies why you gave this tight upper bound. *[65% of the mark for this question]*

Selected Answer:

Worst case running time : Theta(NLog(N)) + Theta(N)
Tight Upper Bound : O(N^2)
Worst-case input of size N: nums=[size N], target = Nth element
Detailed Analysis:
In this algorithm we are sorting all the elements in the beginning, so the worst case scenario here will be the array to be of the maximum size which is of size N. If we go further down into the for loop, we loop it till index < size (of the array) and when the difference is not equal to 0, and in the for loop you have two other variables hi and lo, which start with the last index and index +1 respectively. These two variables are then implemented in a for loop which loops if lo < hi. The worst case here could be that our target is in the middle, this for loop would go on till the hi and lo reach the middle value, which is likely to be subtracted/decreased N/2 times for hi and increased/added N/2 times for lo, which would add up eventually for the complete while loop to iterate over n times. If the target is in the last element, it would take n times for the for loop to iterate and it would also mean that the lo will be incremented to almost N times to reach closer the hi and target. If this happens the for loop iterates n times and the while loop nested in it will also iterates N times, which would go on and give out a tight upper bound of O(N^2).

Correct Answer: [None]

Response Feedback: Deducting marks for:
- analysis does not consider cost of sort clearly.

Question 3

30 out of 30 points

1. Describe in English the intuitive difference between the **big-O** and **big-Theta** notations when used in asymptotic worst-case running time analysis.
2. When might one **prefer** to use the big-Theta notation instead of the big-O notation?

Selected Answer: Big O is gives only upper asymptotic bound, while big Theta also gives a lower bound.
Everything that is Theta(f(n)) is also O(f(n)), but it is not true the other way around.
T(n) is said to be Theta(f(n)), if it is both O(f(n)) and Omega(f(n)),
For this reason Big-Theta is more informative than Big-O notation, so if we can say something is Big-Theta, it's usually preferred.
However, it is harder to prove that something's Big Theta, than it is to prove its big-O.
For example, merge sort is both O(Nlog(N)) and Theta(NLog(N)), but it is also O(N^2), since n^2 is asymptotically "bigger" than it.
However, it is NOT Theta(N^2), Since the algorithm is NOT Omega(n2).

Correct Answer: [None]

Response Feedback: 2- We may want to use the big-Theta notation instead of the big-O notation when we want succinctly express that we know the exact asymptotic order of growth of an algorithm. [Note that the asymptotic notation is independent from the analysis we do (e.g. worst-case analysis).]

Question 4

20 out of 20 points

Problem: you are asked to read a stream of N integers through the standard input (N is large but N integers can be stored in memory), and after reading ends to output the

integers in the same order in the output.

Which of the following Abstract Data Types would be most appropriate to use in the solution of this problem?

Selected Answers:	Queue
Answers:	Max Priority Queue
	Min Priority Queue
	Stack
	Queue

25 June 2021 15:21:57 o'clock BST

← OK