Review Test Submission: Module 1 Assignment

User	Steve Halder
Course	CSC 385 D: Data Structures & Algorithms (Spring 2020)
Test	Module 1 Assignment
Started	1/20/20 9:06 PM
Submitted	1/21/20 4:46 AM LATE
Due Date	1/20/20 11:59 PM
Status	Completed
Attempt Score	13.34 out of 20 points
Time Elapsed	7 hours, 39 minutes
Results Displayed	Submitted Answers

Question 1

1.5 out of 2 points

Match each of the mathematical functions on the left with the corresponding order of magnitude Big-Oh value on the right.

Question	Selected Match
$3 + 4\log N$	D. $O(\log N)$
$5n^3 - 7n + 30$	F. $O(N^3)$
$70000000n^2 + 40000n + 10^8$	C. $O(2^N)$
$(n^3 + 3n^2 + 2n) / 6$	F. $O(N^3)$

Question 2

2 out of 2 points

Assume you have two algorithms, A and B, both of which perform the same function, although their implementations differ. Assume that algorithm A has a running time of $O(N^3 + 2N^2 + 6N + 3)$ and algorithm B has a running time of $O(6N^2 + 4N + 10)$. Also assume that the value of N is restricted to the set of Natural numbers, including 0; i.e., {0, 1, 2, 3, ...}. If the performances of the two algorithms are compared, assuming identical test conditions for both algorithms, match each of the following questions on the left to the correct answer on the right. For this question you must look deeper than the order of magnitude values for Big-Oh.

Question	Selected Match

Which algorithm would be expected to be most efficient when N is arbitrarily large; i.e., as N approaches infinity?	C. Algorithm B
Which algorithm would be expected to be most efficient when N is arbitrarily small; i.e., as N approaches 0?	B. Algorithm A

Question 3

0 out of 2 points

Assume you have two algorithms, A and B, both of which perform the same function, although their implementations differ. Assume that algorithm A has a running time of $O(N^3 + 2N^2 + 6N + 3)$ and algorithm B has a running time of $O(6N^2 + 4N + 10)$. Also assume that the value of N is restricted to the set of Natural numbers, including 0; i.e., {0, 1, 2, 3, ...}. If the performances of the two algorithms are compared, assuming identical test conditions for both algorithms, at what value of N would you expect both algorithms to exhibit approximately the same efficiency?

Selected Answer: 1

Question 4

2 out of 2 points

Determine the precise (i.e., not just the order of magnitude) Big-Oh values for the following code sample, based on the number of statement executions, as described in the Module 1 lecture. Choose the answer that best agrees with the value you have determined. Keep the following considerations in mind:

- Remember to consider each statement in compound statements separately.
- Pay close attention to the initial and end values of loop variables!
- Loops such as "for" and "while" contain an implicit "jump" instruction that causes execution to proceed from the end of the loop back to the beginning of the loop.

Code sample:

```
int i = 0;

int j = 1;
```

Selected Answer: O(2)

Question 5

1.5 out of 2 points

Determine the precise (i.e., not just the order of magnitude) Big-Oh values for the following code sample, based on the number of statement executions, as described in the Module 1 lecture. Choose the answer that best agrees with the value you have determined. Keep the following considerations in mind:

- Remember to consider each statement in compound statements separately.
- Pay close attention to the initial and end values of loop variables!
- Loops such as "for" and "while" contain an implicit "jump" instruction that causes execution to proceed from the end of the loop back to the beginning of the loop.

Code sample:

```
for (int i = 0; i < n; i++)
{
    sum += i;
}
```

Selected Answer: O(3N + 2)

Question 6

1.5 out of 2 points

Determine the precise (i.e., not just the order of magnitude) Big-Oh values for the following code sample, based on the number of statement executions, as described in the Module 1 lecture. Choose the answer that best agrees with the value you have determined. Keep the following considerations in mind:

- Remember to consider each statement in compound statements separately.
- Pay close attention to the initial and end values of loop variables!
- Loops such as "for" and "while" contain an implicit "jump" instruction that causes execution to proceed from the end of the loop back to the beginning of the loop.

Code sample:

```
for (int i = 0; i < n; i++)
{
    sum += i;
}

int j = 0;

while (j < n)
{
    sum--;

    j++;
}
```

Selected Answer: O(6N + 4)

Question 7

1.34 out of 2 points

Determine the precise (i.e., not just the order of magnitude) Big-Oh values for the following code sample, based on the number of statement executions, as described in the Module 1 lecture. Choose the answer that best agrees with the value you have determined. Keep the following considerations in mind:

- Remember to consider each statement in compound statements separately.
- Pay close attention to the initial and end values of loop variables!
- Loops such as "for" and "while" contain an implicit "jump" instruction that causes execution to proceed from the end of the loop back to the beginning of the loop.

Code sample:

```
for (int i = 0; i < n; i++)
{
    for (int j = i; j < n; j++)
    {
        sum += i;
    }
}
```

Selected Answer: $O(4N^2 + 5N + 2)$

Question 8

2 out of 2 points

Determine the precise (i.e., not just the order of magnitude) Big-Oh values for the following code sample, based on the number of statement executions, as described in the Module 1 lecture. Choose the answer that best agrees with the value you have determined. Keep the following considerations in mind:

- Remember to consider each statement in compound statements separately.
- Pay close attention to the initial and end values of loop variables!
- Loops such as "for" and "while" contain an implicit "jump" instruction that causes execution to proceed from the end of the loop back to the beginning of the loop.

Code sample:

```
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        sum += i;
    }
}
```

Selected Answer: $O(4N^2 + 5N + 2)$

Question 9

0 out of 2 points

Determine the precise (i.e., not just the order of magnitude) Big-Oh values for the following code sample, based on the number of statement executions, as described in the Module 1 lecture. Choose the answer that best agrees with the value you have determined. Keep the following considerations in mind:

- Remember to consider each statement in compound statements separately.
- Pay close attention to the initial and end values of loop variables!
- Loops such as "for" and "while" contain an implicit "jump" instruction that causes execution to proceed from the end of the loop back to the beginning of the loop.

Code sample:

```
for (int i = 0; i < n; i++)
{
    sum += i;
}
```

```
for (int j = 0; j < n; j++)
{
    for (int k = 0; k < n; k++)
    {
        sum--;
    }
}
```

Selected Answer: $O(16N^3 + 28N^2 + 18N + 4)$

Question 10

1.5 out of 2 points

Determine the precise (i.e., not just the order of magnitude) Big-Oh values for the following code sample, based on the number of statement executions, as described in the Module 1 lecture. Choose the answer that best agrees with the value you have determined. Keep the following considerations in mind:

- Remember to consider each statement in compound statements separately.
- Pay close attention to the initial and end values of loop variables!
- Loops such as "for" and "while" contain an implicit "jump" instruction that causes execution to proceed from the end of the loop back to the beginning of the loop.

Code sample:

```
for (int i = 0; i < n; i++)
{
    for (int j = i; j < n; j++)
    {
        for (int k = j; k < n; k++)
        {
            sum += j;
        }
    }
}
```

Selected Answer: $O((3N^3 + 21N^2 + 42N + 12)/6)$

Tuesday, May 12, 2020 10:52:48 AM CDT

← OK