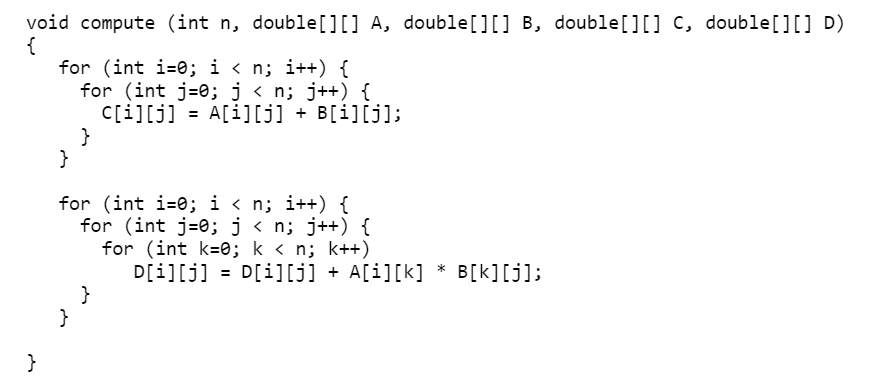
Anders Lykkehoy

10/14/2018

CMPT 435

Assignment 2

1. Analyze the following code and provide a "Big-O" estimate of its running time in terms of n. Explain your analysis.



Note: Credit will not be given only for answers - show all your work:

(5 points) steps you took to get your answer.

The first part of the function starts with two nested for loops. This gives us a big O for that section of n(n) or n^2. The second part of the function has three nested for loops. The big O of that is n(n(n))) or n^3. This means combined we have n^3 + n^2. Since the n^3 is larger than n^2, our overall big O for this function is O(n^3).

(5 points) your answer.

The big O of this function is O(n^3).

1. Analyze the following code and provide a "Big-O" estimate of its running time in terms of n. Explain your analysis.

int j = 1, i = 0;

while (i < n)

{

i = i + j;

j++;

}

**Hint: The loop variable ‘i’ is incremented by 1, 2, 3, 4, … until i becomes greater than or equal to n.**

Note: Credit will not be given only for answers - show all your work:

(5 points) steps you took to get your answer.

To find the big O of this we need to look at the while loop. Since we know j increases by one every loop and i increases by j, i will be 0, 1, 3, 6, 10, 15, 21… up to or passing n. Therefore we know the big O of this code is O(log n).

(5 points) your answer.

The big O of this code is O(log n).