CIS 263 Assignment 2

Jack Lukomski

Dr. Bobekdyk

6/29/2023

**Question 1:**

for (int i = 0; i < n; i++) { // n + 1

total++; // n

}

Answer:

**Code Analysis:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n | clock ticks | Question 1 | | | | | |
| 1 | 4 |
| 2 | 2 |  |  |  |  |  |  |
| 4 | 2 | |  | | --- | |  | |  |  |  |  |  |
| 8 | 3 |  |  |  |  |  |  |
| 16 | 2 |  |  |  |  |  |  |
| 32 | 2 |  |  |  |  |  |  |
| 64 | 2 |  |  |  |  |  |  |
| 128 | 2 |  |  |  |  |  |  |
| 256 | 3 |  |  |  |  |  |  |
| 512 | 5 |  |  |  |  |  |  |
| 1024 | 5 |  |  |  |  |  |  |
| 2048 | 10 |  |  |  |  |  |  |
| 4096 | 7 |  |  |  |  |  |  |
| 8192 | 14 |  |  |  |  |  |  |
| 16384 | 64 |  |  |  |  |  |  |
| 32768 | 65 |  |  |  |  |  |  |
| 65536 | 97 |  |  |  |  |  |  |
| 131072 | 620 |  |  |  |  |  |  |
| 262144 | 558 |  |  |  |  |  |  |
| 524288 | 832 |  |  |  |  |  |  |

As you can see, when n takes on the values from 0 – 524,288, the number of clock ticks in execution of the program follows a relative linear line. Obviously, this is not a perfect linear line but as , this line will fall into a more linear fashion, as predicted.

**Question 2:**

for (int i = 0; i < n; i++) { // n + 1

for (int j = 0; j < n; j++) { // (n + 1) \* n

total++; // n \* n

}

}

Answer:

**Code Analysis:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n | clock ticks | Question 2 | | | | | |
| 1 | 3 |
| 2 | 1 |  |  |  |  |  |  |
| 4 | 2 | |  | | --- | |  | |  |  |  |  |  |
| 8 | 2 |  |  |  |  |  |  |
| 16 | 3 |  |  |  |  |  |  |
| 32 | 2 |  |  |  |  |  |  |
| 64 | 8 |  |  |  |  |  |  |
| 128 | 22 |  |  |  |  |  |  |
| 256 | 80 |  |  |  |  |  |  |
| 512 | 303 |  |  |  |  |  |  |
| 1024 | 1196 |  |  |  |  |  |  |
| 2048 | 4776 |  |  |  |  |  |  |
| 4096 | 15210 |  |  |  |  |  |  |
| 8192 | 43867 |  |  |  |  |  |  |
| 16384 | 170193 |  |  |  |  |  |  |
| 32768 | 737298 |  |  |  |  |  |  |
| 65536 | 3039558 |  |  |  |  |  |  |
| 131072 | 14825380 |  |  |  |  |  |  |
| 262144 | 55061556 |  |  |  |  |  |  |
| 524288 | 211682141 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

As you can see, as n gets larger, the form of the line takes on an exponential path (). This follows our prediction of the Big O for this function.

**Question 3:**

for (int i = 0; i < n; i++) { // n + 1

for (j = 0; j < i; j++) { // i + 1

total++; // n \* i (where i can be 1, 2, 3, . . . n)

}

}

Worst case run time would be when , thus making this the same problem as in question 2.

Answer:

**Code Analysis:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n | clock ticks | Question 3 | | | | | |
| 1 | 5 |
| 2 | 3 |  |  |  |  |  |  |
| 4 | 2 |  |  |  |  |  |  |
| 8 | 2 | |  | | --- | |  | |  |  |  |  |  |
| 16 | 3 |  |  |  |  |  |  |
| 32 | 5 |  |  |  |  |  |  |
| 64 | 8 |  |  |  |  |  |  |
| 128 | 23 |  |  |  |  |  |  |
| 256 | 70 |  |  |  |  |  |  |
| 512 | 251 |  |  |  |  |  |  |
| 1024 | 894 |  |  |  |  |  |  |
| 2048 | 3428 |  |  |  |  |  |  |
| 4096 | 10372 |  |  |  |  |  |  |
| 8192 | 38024 |  |  |  |  |  |  |
| 16384 | 92420 |  |  |  |  |  |  |
| 32768 | 346734 |  |  |  |  |  |  |
| 65536 | 1487307 |  |  |  |  |  |  |
| 131072 | 5954922 |  |  |  |  |  |  |
| 262144 | 25004307 |  |  |  |  |  |  |
| 524288 | 101602238 |  |  |  |  |  |  |

As you can see, as n gets larger, the form of the line takes on an exponential path (). Also, the number of clock ticks for is less than that of question 2, and both functions have the same Big O. This is because the second level for loop is being executed less in this problem than in question 2. In question 2, the second level for loop goes from 0 to n each call, but in this problem, the second level for loop goes from 0 to i, meaning that it goes from 0 – n only one time.

**Question 4:**

for (int i = 0; i < n; i++) { // n + 1

for (int j = 0; j < i; j++) { // i + 1

if (j % 2 == 0) { // n \* i (where i can be 1, 2, 3, . . . n)

total++; // (n \* i)/2 \* (n \* i)/2

}

}

}

This function’s Big O is still because assuming the worst case, , the function will still take quadratic time.

**Code Analysis:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n | clock ticks | Question 4 | | | | | |
| 1 | 3 |
| 2 | 2 |  |  |  |  |  |  |
| 4 | 2 |  |  |  |  |  |  |
| 8 | 2 | |  | | --- | |  | |  |  |  |  |  |
| 16 | 3 |  |  |  |  |  |  |
| 32 | 4 |  |  |  |  |  |  |
| 64 | 8 |  |  |  |  |  |  |
| 128 | 30 |  |  |  |  |  |  |
| 256 | 113 |  |  |  |  |  |  |
| 512 | 440 |  |  |  |  |  |  |
| 1024 | 1765 |  |  |  |  |  |  |
| 2048 | 7020 |  |  |  |  |  |  |
| 4096 | 28583 |  |  |  |  |  |  |
| 8192 | 71493 |  |  |  |  |  |  |
| 16384 | 251829 |  |  |  |  |  |  |
| 32768 | 1007589 |  |  |  |  |  |  |
| 65536 | 3994197 |  |  |  |  |  |  |
| 131072 | 16247458 |  |  |  |  |  |  |
| 262144 | 90242056 |  |  |  |  |  |  |
| 524288 | 305570905 |  |  |  |  |  |  |

As you can see, as n gets larger, the form of the line takes on an exponential path (). Also, this function has the same loop style as question 3 but takes significantly longer to execute. This is because inside the for loops there is a conditional that the program also has to check each pass through.