Matthew Ricketts
Homework 1
2/7/2022

For each of the following pairs of functions f (n) and g(n), state whether f (n) = O(g(n)), f (n) = Ω(g(n)), or neither.

1. f (n) = $n^2 + 42n - 137$, g(n) = 6n + 7

   $n^2 > n$, f(n) = O(g(n))

2. f (n) = $n^{3/2}$, g(n) = $8n^2 - 3n$

   $n^{3/2} < n^2$, f(n) = Ω(g(n))

3. f (n) = $2n - n^2$, g(n) = $(n^4 + n^2 + 8)/n$

   $n < n^3$, f(n) = Ω(g(n))

An algorithm takes 0.5 ms for input size 100. How long will it take for input size 500 if the running time is the following (assume low-order terms are negligible)?

a. linear

$$0.5 * \frac{500}{100} = 2.5\ seconds$$

b. O(N logN)

$$0.5 * \frac{500\log(500)}{100\log(100)} = 3.37\ seconds$$

c. Quadratic

$$0.5 * \frac{500^2}{100^2} = 12.5\ seconds$$

d. Cubic

$$0.5 * \frac{500^3}{100^3} = 62.5\ seconds$$

An algorithm takes 0.5 ms for input size 100. How large a problem can be solved in 1 min if the running time is the following (assume low-order terms are negligible)?

$$time * \frac{{size_{theoretical}}^{fact}}{{size_{known}}^{fact}} = time_{total}$$

$${size_{theoretical}}^{fact} = {size_{known}}^{fact} * \frac{time_{total}}{time}$$

1 min = 60,000 seconds

a. linear

$$100^1 * \frac{60{,}000}{0.5} = 12{,}000{,}000$$

$$size_{theoretical} = 3{,}656{,}807$$

b. O(N logN)

Matthew Ricketts
Homework 1
2/7/2022

$$100\log(100) * \frac{60{,}000}{0.5} = size_{theoretical}\log(size_{theoretical})$$

$$size_{theoretical} = 3{,}656{,}807$$

c. Quadratic

$$100^2 * \frac{60{,}000}{0.5} = size_{theoretical}{}^2$$

$$size_{theoretical} = 34{,}641$$

For each of the following six program fragments:

a. Give an analysis of the running time (Big-Oh will do).

b. Implement the code in the language of your choice, and give the running time for severalvalues of N. (in this case running time = sum)

c. Compare your analysis with the actual running times. (Short answer)

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

// 2
for (int i = 0; i < n; ++i)
        for (int j = 0; j < n;++j)
                ++sum;
// 3
for (int i = 0; i < n; ++i)
        for (int j = 0; j < n*n; ++j)
                ++sum;
// 4
for (int i = 0; i < n; ++i)
        for (int j = 0; j < i; ++j)
                ++sum;

// 5
for (int i = 0; i < n; ++i)
        for (int j = 0; j < i*i; ++j)
                for (int k=0;k<j;k++)
                        ++sum;

// 6
sum = 0;
for (int i = 0; i < n; ++i)
        for (int j = 0; j < i * i; ++j)
                if (j % i == 0)
                        for (int k = 0; k < j; k++)
                                ++sum;
```

Analysis

1. One loop to n
   O(n)
2. Loop to n nested in loop to n
   O(n*n) = O(n^2)
3. Loop to n*n nested in loop to n

Matthew Ricketts
Homework 1
2/7/2022

$O((n*n) * n) = O(n^3)$

4. i can be as big as n
   Loop to i nested in loop to n
   $O(i*n) = O(n*n) = O(n^2)$
5. i can be as big as n
   j can be as big as i*i
   loop to j nested in loop to i*i nested in loop to n
   $O(n*i*i*j) = O(n*i*i*i*i) = O(n*n*n*n*n) = O(n^5)$
6. i can be as big as n
   j can be as big as i*i
   j % i statement makes k to j loop execute every j/i loops
   loop to j/i nested in loop to i*i nested in loop to n
   $O(n*i*i*j) = O(n*i*i*j/i) = O(n*i*i*i*i/i) = O(n*i*i*i) = O(n*n*n*n) = O(n^4)$

Code Implementation Run Time Results

N = 10

Test 1: sum = 10
Test 2: sum = 100
Test 3: sum = 1000
Test 4: sum = 45
Test 5: sum = 7524
Test 6: sum = 870

N = 100

Test 1: sum = 100
Test 2: sum = 10000
Test 3: sum = 1000000
Test 4: sum = 4950
Test 5: sum = 975002490
Test 6: sum = 12087075

Analysis to Run Time Comparison

N = 10

Test 1: sum = 10
  $10 = 10^1$, therefore $O(n)$ is correct
Test 2: sum = 100
  $100 = 10^2$, therefore $O(n^2)$ is correct
Test 3: sum = 1000
  $1000 = 10^3$, therefore $O(n^3)$ is correct
Test 4: sum = 45
  $10\log(10) < 45 < 10^2$, therefore $O(n^2)$ is correct
Test 5: sum = 7524
  $7524 < 10^4$, not 10^5 is $O(n^5)$ is correct????
Test 6: sum = 870
  $870 < 10^3$, not 10^4 is $O(n^4)$ is correct????

N = 100

Matthew Ricketts
Homework 1
2/7/2022

Test 1: sum = 100
  $100 = 100^1$, therefore $O(n)$ is correct
Test 2: sum = 10000
  $10000 = 100^2$, therefore $O(n^2)$ is correct
Test 3: sum = 1000000
  $1000000 = 100^3$, therefore $O(n^3)$ is correct
Test 4: sum = 4950
  $100\log(100) < 4950 < 10^2$, therefore $O(n^2)$ is correct
Test 5: sum = 975002490
  $100^4\log(100) < 975002490 < 10^5$, therefore $O(n^5)$ is correct
Test 6: sum = 12087075
  $100^3\log(100) < 12087075 < 10^4$, therefore $O(n^4)$ is correct

Conclusion: as N grows, the below results are correct

1. $O(n)$
2. $O(n^2)$
3. $O(n^3)$
4. $O(n^2)$
5. $O(n^5)$
6. $O(n^4)$