CS 1181 Project 4

By:

Cameron Windsor Fox

In this project we are given code fragments we must use these fragments and find the Big-O by looking at each fragment bit by bit. Then find the expected Growth factor by double each Big-O value you used in each fragment. Then we run the code and get the Observed Growth Factor. We first must look at the code fragments we are given and then determine the growth after the code runs.

|  |  |  |  |
| --- | --- | --- | --- |
| Fragment # | Big-O | Expected Growth Factor | Observed Growth Factor |
| 1 | O(n) | 2n | 2.009 |
| 2 | O(n^2) | 2n^2 | 4.003 |
| 3 | O(2n) | 4n | 2.004 |
| 4 | O(n^3) | 2n^3 | 7.946 |
| 5 | O(n^2) | 2n^2 | 3.974 |
| 6 | O(n^5) | 2n^5 | 28.530 |
| 7 | O(log n) | 1 + log n | 1.058 |
| 8 | O(2^n) | 2^2n | 3.999 |

For each code fragment there is a specific growth factor that had to be founded. The first fragment used an expected growth factor of 2n because there is one for loop used equaling a growth factor of O (n). The second fragment growth is 2n^2 because this fragment uses a nested for loop so the Big-O is n^2. For the third fragment this is double of the first fragment so the Big-O is 2n so the growth factor will be 4n. The fourth fragment the Big-O is n^3 because you have a normal for loop and another for loop that uses a squared value, so n \* n^2 is going to be n^3. With that the growth will be 2n^3.

The fifth fragment the Big-O is just n^2 because you just have a nested for loop. So then the growth factor will be 2n^2. The sixth fragment has a Big-O of n^5 because you have two for loops using a squared n ad then a normal for loop. So that is n, n^2, n^2 together that is n^5. Then the growth factor is 2n^5. The seventh fragment has a Big-O of O (log n). This time there is a while loop being used instead of a for loop. The growth factor of this is going to be 1 + log n. Finally, the eighth fragment has a Big-O of 2^n because this fragment is using math.power different from all the other fragments. The growth factor of this will be 2^2n.

The connection between the expected growth and the observed growth factor is that the expected shows how long the program should take and the resulting number for the observed growth is around how long the program look to run. Comparing them shows how close we were to guess how long the program will take to run each code fragment. This is how the growth rate of each fragment can be compared.