**Assignment 2**

(1) The estimate running time (or memory) is a function of input size N. Explain as to why the

results are the same for the following three examples.

Ans:

* T(N) = 1/6 N^3 + 20N + 16:

by the Big-Oh definition, T(N) is O (N^3) if T(N) ≤ c \* N^3

for some N ≥ N0, if 1/6 N^3 + 20N + 16 ≤ c\*N^3 and if we move N^3 on other side of the equation the below will be the new form of equation.

1/6 + 20/N^2 + 16/N^3 ≤ c. Therefore, the big-oh condition for N will be N ≥ N0=1 and c ≥ 36.16 (1/6+20+16) which means as we increase the value of N, value of c getting decrease due to smaller factors. That’s why the 20N + 16 is negligible compared to 1/6 N^3 term.

For this reason, running time for T(N) is ~1/6 N^3.

* T(N) = 1/6 N^3 + 100N^4/3 + 56:

by the Big-Oh definition, T(N) is O (N^3) if T(N) ≤ c \* N^3

for some N ≥ N0, if 1/6 N^3 + 100N^4/3 + 56 ≤ c\*N^3 and if we move N^3 on other side of the equation the below will be the new form of equation.

1/6 + 100/N^5/4 + 56/N^3 ≤ c. Therefore, the big-oh condition for N will be N ≥ N0=1 and c ≥ 156.16 (1/6+100+56) which means as we increase the value of N, value of c getting decrease due to smaller factors. Value of c is more compared than first T(N) function but still it is not significantly big. That’s why the 100N^4/3 + 56 is negligible compared to 1/6 N^3 term.

For this reason, running time for T(N) is ~1/6 N^3.

* T(N) = 1/6 N^3 – 1/2N^2 + 1/3N:

by the Big-Oh definition, T(N) is O (N^3) if T(N) ≤ c \* N^3

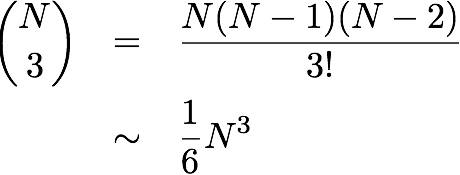
for some N ≥ N0, if 1/6 N^3 – 1/2N^2 + 1/3N ≤ c\*N^3 and if we move N^3 on other side of the equation the below will be the new form of equation.

1/6 -1/2N + 1/3N^2 ≤ c. Therefore, the big-oh condition for N will be N ≥ N0=1 and c ≥ 0 (1/6-1/2+1/3) which means as we increase the value of N, value of c getting decrease because second and third terms are not significant since higher value of N doesn’t contribute much compared to first term(1/6 N^3). That’s why the – 1/2N^2 + 1/3N is negligible compared to 1/6 N^3 term.

For this reason, running time for T(N) is ~1/6 N^3.

(2) Write the Java code samples for the running times: constant 1, logN, N, NlogN, N^2,

N^3, 2^N. Mathematically, how do you describe each one of these examples in the form of

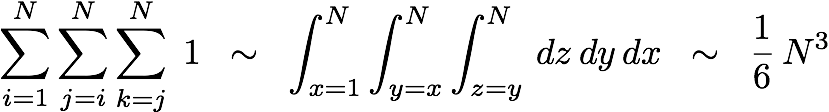
following equation?

Ans: Please find table mention below for each example in require form.

|  |  |  |
| --- | --- | --- |
| Order of Growth | Framework | Example |
| 1 | a = b +c | Add two number |
| logN | While(N>1){N = N/2;..} | Binary search |
| N | for (int i = 0; i < N; i++) { ... } | Find the maximum |
| NlogN | For(int i=0;i<=N;i=i\*2)  For(int j=0;j<N;j++)  { .. } | Merge sort |
| N^2 | For(int i=0;i<=N;i=i++)  For(int j=0;j<N;j++)  { .. } | Check all pair |
| N^3 | For(int i=0;i<=N;i=i++)  For(int j=0;j<N;j++)  For(int k=0;k<N;k++)  { .. } | Check all triple pairs |
| 2^N | if (num <= 1) return num;  return fibonacci(num - 2) + fibonacci(num - 1); | Fibonacci series |

(3) Write the code that results to following running time. The 3-Sum Triple loop has the following

running time estimate. Do Not prove the math. Just want to explain the math, what it

represents and why the result is 1/6 N^3

Ans:

Integration and summation are very similar. For 3-sum triple loop, out of N elements of array, only 3 elements of array will access for condition in loop. This means that combination of 3 element of array will be selected out of N elements which represents as nC3.

nC3 = n(n-1) (n-2)/3!

~ 1/6 n^3

(4) Human use Infix expression and computers use Postfix expression. You are to write a simple Calculator. There are three steps: a) Read Infix expression, b) Convert Infix expression to Postfix by hand, and c) Evaluate Postfix expression, d) Use the referenced c-program example and write it in Java code, compile and run with four Infix expression examples.

(1 + 2 \* (20 / 5 ))

(1 + 3 + ( ( 4 / 2 ) \* ( 8 \* 4 ) ))

(4 + 8) \* (6 - 5)/((3 - 2) \* (2 + 2))

(1 + 2 - 3)

Ans: Please check pdf attached with screenshots for all the infix and postfix evaluation.

Please check java files for java code for simple calculator.