Problem Statement

We are required to analyze the following program/code sample.

```
for (int i = 1; i <= n; i++) {
  for (int j = i; j <= n; j++) {
    for (int k = j; k <= n; k++) {
      Sum += a[i] * b[j] * c[k];]
    if (j == 2 * i) {
      j = n; }}}
```

Theoretical Analysis

- The outermost loop (for i) iterates n times.
- The middle loop (for j) iterates for each i from i to n, which is approximately n - i + 1 times for each i.
- The innermost loop (for k) iterates for each j from j to n, which is approximately n - j + 1 times for each j.
- Total Time Complexity = O(n) * O(n - i + 1) * O(n - j + 1)

Therefore, the time complexity of code snippet is O(n^3).

Experimental Analysis

Program Listing

I have executed the code for values of "n" ranging from 10 to 200 with increment of 10 every iteration.

Data Normalization Notes

I converted the experimental values into nanoseconds, while the theoretical values were expressed as large numerical figures. To make a meaningful comparison, I computed the average for both the experimental and theoretical values and subsequently divided them. The outcome provided a scaling factor, which facilitated the comparison between the two sets of data.

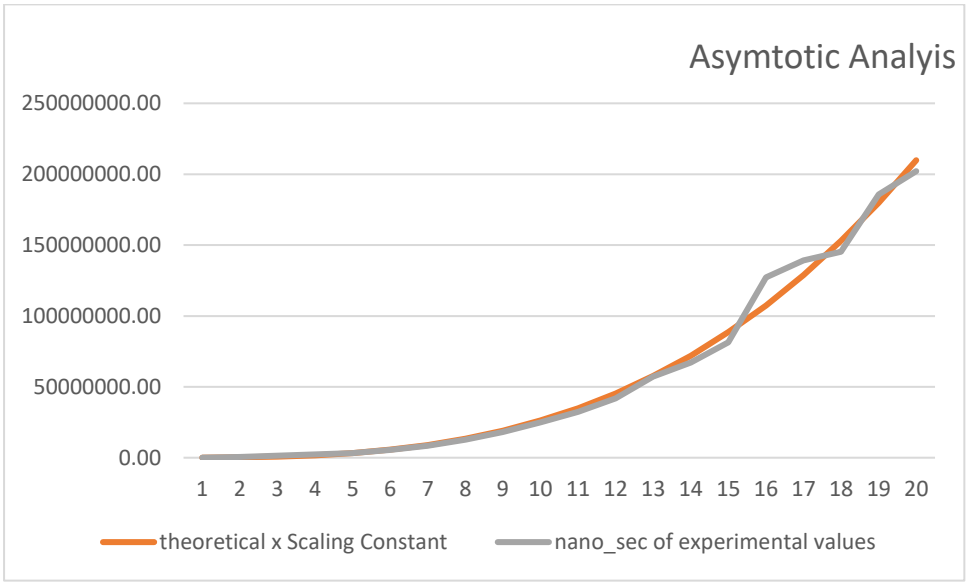
Scaling Factor - 26.23

Output Table

Value of 'n'	Theoretical Time Taken (O(n^3)	Experimental Time Taken	Scaling Constant	theoretical x Scaling Constant	nanosecond of experimental values
1.00E+01	1000.00	0.000041	2.62E+01	26230.66	41000
2.00E+01	8000.00	0.000431	2.62E+01	209845.26	431000
3.00E+01	27000.00	0.001363	2.62E+01	708227.76	1363000
4.00E+01	64000.00	0.002394	2.62E+01	1678762.09	2394000
5.00E+01	125000.00	0.00326	2.62E+01	3278832.20	3260000
6.00E+01	216000.00	0.00548	2.62E+01	5665822.04	5480000
7.00E+01	343000.00	0.008537	2.62E+01	8997115.56	8537000
8.00E+01	512000.00	0.012712	2.62E+01	13430096.69	12712000
9.00E+01	729000.00	0.018141	2.62E+01	19122149.39	18141000
1.00E+02	1000000.00	0.024939	2.62E+01	26230657.60	24939000
1.10E+02	1331000.00	0.03248	2.62E+01	34913005.26	32480000

1.20E+02	1728000.00	0.041795	2.62E+01	45326576.33	41795000
1.30E+02	2197000.00	0.057218	2.62E+01	57628754.74	57218000
1.40E+02	2744000.00	0.067141	2.62E+01	71976924.44	67141000
1.50E+02	3375000.00	0.0813	2.62E+01	88528469.39	81300000
1.60E+02	4096000.00	0.127208	2.62E+01	107440773.51	127208000
1.70E+02	4913000.00	0.13916	2.62E+01	128871220.77	139160000
1.80E+02	5832000.00	0.145405	2.62E+01	152977195.10	145405000
1.90E+02	6859000.00	0.185612	2.62E+01	179916080.45	185612000
2.00E+02	8000000.00	0.202155	2.62E+01	209845260.77	202155000

Graph



Analysis:

We can visually derive from the graph that the points are generally aligned with the theoretical line. This means that the theoretical scalling constant is good approximation of the runtime of algorithm.

Due to factors like noise and variations in the implementation of algorithm we can see few lines fall below the line

Overall, the graph shows the asymptotic analysis of the algorithm is accurate.

Results:

We can conclude that Theoretical time complexity of O(n^3) is correct, which can be validated from the results.