Project 1

# Problem Statement

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

int j = 2

while (j < n) {

int k = j

while (k < n) {

Sum += a[k]\*b[k]

k += n1/3 log n

}

j = 2\*j

}

# Theoretical Analysis

There are 2 while loops in the above mentioned code. If we calculate the time complexity of the inner loop, we will find that the increment variable of the loop (k) is being increased by the increment of n1/3 log n*.* And we also see that the loop has a stopping condition of n, means that when k = n, the loop will stop. So the loop will run for n/ n1/3 log n times. Similarly, The outer loop will run for log n times as j is being incremented with a product of 2. As the 2 loops are nested, we multiply the complexity. So (n/ n1/3 log n) \* log n = n2/3 .

So the time complexity of the program is O (n2/3).

# Experimental Analysis

## Program Listing

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

#include<time.h>

Void main()

{clrscr();

Time\_t begin=time(NULL);

int j = 2

while (j < n) {

int k = j

while (k < n) {

Sum += a[k]\*b[k]

k += n1/3 log n

}

j = 2\*j

}  
Time\_t stop=time(NULL);

cout<<”Execution Time:”<<(double)(stop-begin);

getch();

}

We will run this program, first continuously and then independently for n=10, 100, 1000, 10000, 100000, 1000000, 10000000. Not taking n=1 because control won’t enter the loop as j=2.

## Data Normalization Notes

To normalize the theoretical values obtained and to compare the actual experimental results, we will have to multiply the theoretical values with 100 as the values are in nanoseconds.

## Output Numerical Data

|  |  |  |  |
| --- | --- | --- | --- |
| N | Experimental (ns) | Theoretical (ns) | Adjusted Theoretical Value (\*100) |
| 10 | 657 | 4.64 | 464 |
| 100 | 2300 | 21.54 | 2154 |
| 1,000 | 8260 | 100 | 10000 |
| 10,000 | 37890 | 464.20 | 46420 |
| 100,000 | 182200 | 2154.50 | 215450 |
| 1,000,000 | 815500 | 10000 | 1000000 |
| 10,000,000 | 4352200 | 46415.90 | 4641589 |

## Graph

## Graph Observations

The graph above shows that both the lines coincide very well but there are some value of n where the lines do not coincide. This shows that even though, the theoretical values seem very fitting, the experimental values still have some sort of deviation. But the overall hue of both the graph lines indicate that the time complexity calculated must be similar to that which was calculated.

# Conclusions

The analysis of the algorithm as O (n2/3) seems to be supporting the graph. May be a better analysis of the scatterplots can be revealed if we change programming environments, but this graph seems to give the user a good idea that the complexity is, nevertheless, is O (n2/3).