Project 1

Author Name
Qi Zhao （GWID:G41261092）

1 Problem Statement

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

```
int j = 2
while (j < n) {
  int k = 2
  while (k < n) {
    Sum += a[k]*b[k]
    K = k * sqrt(k);
  }
  j+=j/2;
}
```

2 Theoretical Analysis

Explain your theoretical estimate in 3-4 sentences.

Reasoning: For the inner loop, the number of iteration t1: $2^{\frac{3^{t1}}{2}} \sim n$, *so* $t1 \sim loglogn$

   For the outer loop, the number of iteration t2: $\frac{3^{t2}}{2} * 2 \sim n$, *so t2~logn*

   Such that time complexity equals O(t1*t2)=O(log(n)*loglog(n));

Mathematical expressions. O(logn*loglogn);

3 Experimental Analysis

3.1 Program Listing

(Feel free to include only selected portions if you like. For example, I would like to know which values of "n" you ran the program for.)
n :
2^1 2^2 2^3 2^4 2^5 2^6 2^7 2^8 2^9 2^{10} 2^{11} 2^{12} 2^{13} 2^{14} 2^{15} 2^{16} 2^{17} 2^{18} 2^{19} 2^{20} 2^{21} 2^{22} 2^{23} 2^{24} 2^{25}
2^{26} 2^{27} 2^{28} 2^{29} 2^{30} 2^{31} 2^{32} 2^{33} 2^{34} 2^{35} 2^{36} 2^{37} 2^{38} 2^{39} 2^{40} 2^{41} 2^{42} 2^{43} 2^{44} 2^{45} 2^{46} 2^{47} 2^{48} 2^{49} 2^{50}

3.2 Data Normalization Notes

Do you normalize the values by some constant? How did you derive that constant?
Yes, I compute the theoretical run time by multiplying time complexity by a constant.
The constant=(experimental run time(n1)-experimental run time(n2))/(time complexity(n1)-time complexity(n2))

3.3   Output Numerical Data

N	Experimental(ns)	Theoretical(ns)
8192	10225	7179
1.34*10^8	25035	22897
5.63*10^14	56419	53373

3.4   Graph
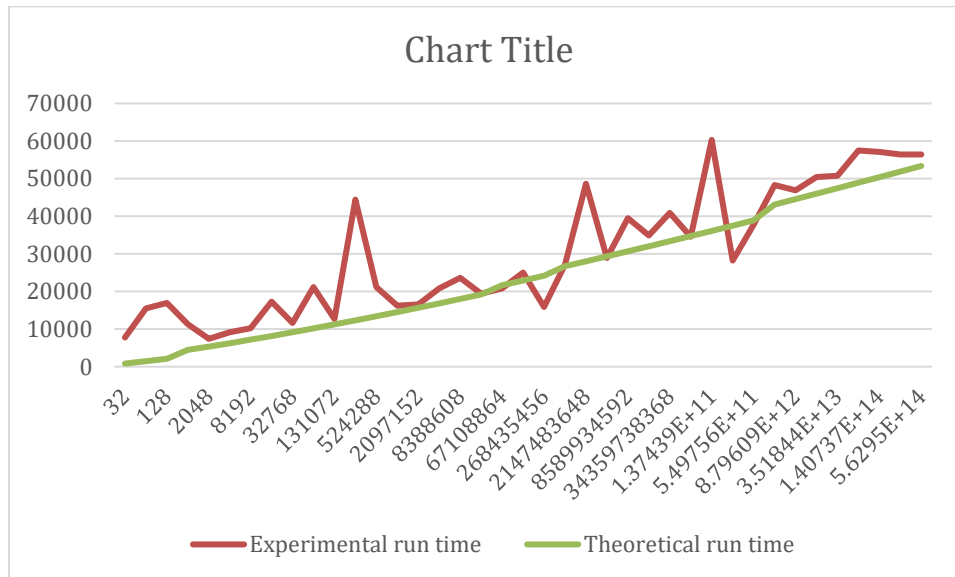
3.5   Graph Observations

The tendency and rising rate of the experimental run time is approximately equal to the theoretical run time. There are many big waves in the line of Experimental run time, because the run time is very little and it is easily influenced by the factors of outer environment.

4   Conclusions

The run time of this algorithm is approximately proportional to the time complexity calculated above. So the calculation is right.