Theoretical Part

Runtime Comparison

To compare, LibreOffice Calc was used. A range of 1 - 15 with 15 being the max nth value was used and plugged into each function. The list from lowest runtime to highest runtime after 15 was plugged is as shown.

Lowest to Highest

1) 2 / N
2) N * log(N)
3) log(N)
4) sqrt(N)
5) N
6) N * log(N)
7) N * log^2(n)
8) N * log(N^2)
9) 37
10) N^1.5
11) N^2
12) N^2 * log(N)
13) N^3
14) 2^N

It is important to note, however, that while this test was done on a limited nth, if n -> infinity, the constant value of 37 would change from 9th on the list to 2nd, as every other function besides 2 / N would approach infinity on the y-axis while 2 / N approaches 0 and 37 remains constant.

Some functions contain a similar growth rate. For example:

- N * log(N) and N * log^2(N)
- N^2 and N^2 * log(N)
- log(N) and N * log(log(N))

Asymptotic Notations

1)

```
sum = 0;
for(i = 0; i < n; i++) {
   sum++;
}
```

ANSWER: O(1 + (n + 1)) = O(n)

2)

```
sum = 0;
for(i = 0; i < n; i++) {
    for(j = 0; j < n; j++) {
        sum++;
    }
}
```

ANSWER: O(1 + (n * n)) = O(n^2)

3)

```
sum = 0;
for(i = 0; i < n; i++) {
    for(j = 0; j < n * n; j++) {
        sum++;
    }
}
```

ANSWER: O(1 + (n * n^2)) = O(n^3)

4)

```
sum = 0;
for(i = 0; i < n; i++) {
    for(j = 0; j < i; j++) {
        sum++;
    }
}
```

ANSWER: O(1 + (n * n-1)) = O(n^2)

5)

```
sum = 0;
for(i = 0; i < n; i++) {
    for(j = 0; j < i; j++) {
        for(k = 0; k < j; k++) {
            sum++;
        }
    }
}
```

ANSWER: O(1 + (n * n-1 * n-2)) = O(n^2)

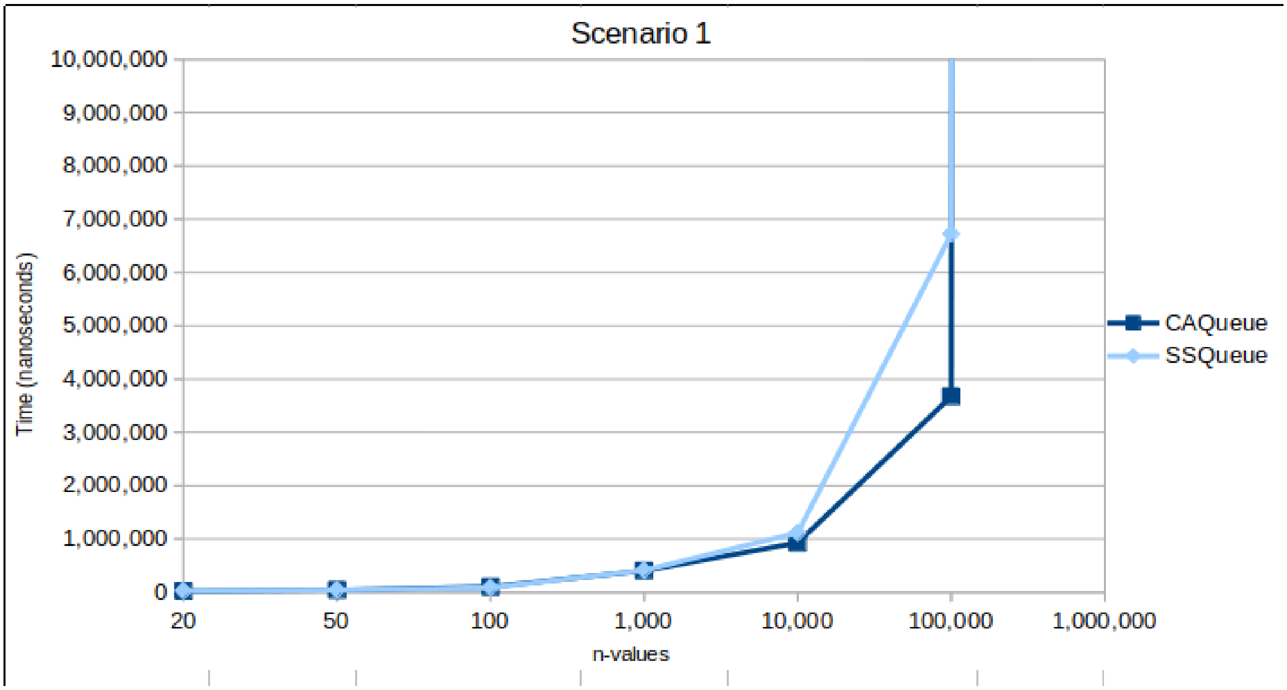
6)

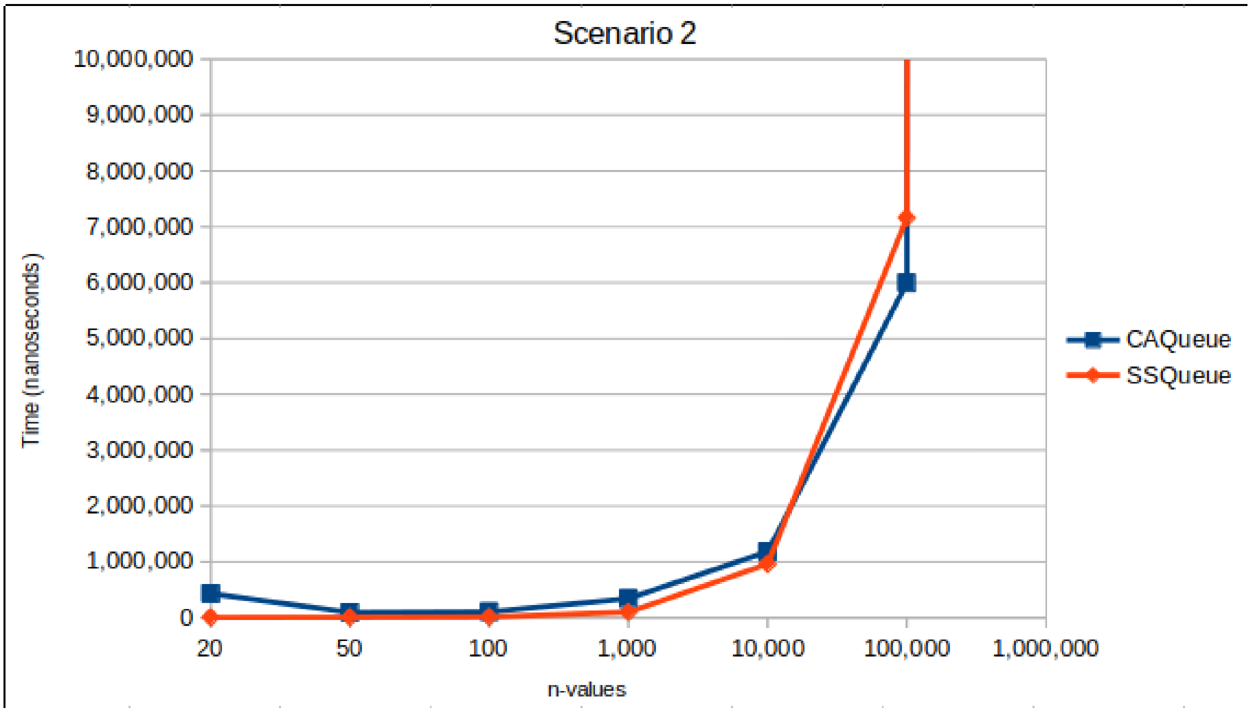
```
sum = 0;
for(i = 1; i < n; i++) {
    for(j = 1; j < i * i; j++) {
        if(j % i == 0) {
            for(k = 0; k < j; k++) {
                sum++;
            }
        }
    }
}
```

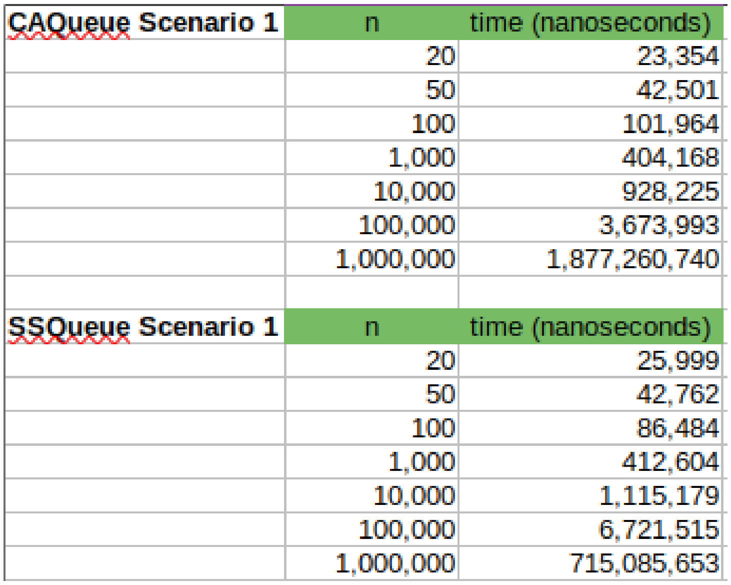
ANSWER: O(1 + (n-1 * n^2 * 1 * n / 2)) = O(n^4)

Programming Part

Queues and Stacks Data Graph (Extra Credit)







CAQueue Scenario 1	n	time (nanoseconds)
	20	23,354
	50	42,501
	100	101,964
	1,000	404,168
	10,000	928,225
	100,000	3,673,993
	1,000,000	1,877,260,740
SSQueue Scenario 1	n	time (nanoseconds)
	20	25,999
	50	42,762
	100	86,484
	1,000	412,604
	10,000	1,115,179
	100,000	6,721,515
	1,000,000	715,085,653