**Braden Morrison (430002127) Lab 1**

Problem 1

Foo: 221Nlog2N Bar: 2N2

(1): If N> 10000 then Foo will take less time than Bar

(2): If N < 100 then Bar will take less time than Foo.

(3): No, once the numbers get large enough (larger than n0), Foo will start to take less time.

Chart

Description automatically generated

Problem 2

Fragment 1:

(a). O(n2)

(b). \*attached to submission on canvas

(c). the runtime for both fragments grew considerably as the input size increased. When the input size was doubled, the runtime was quadrupled, this is congruent with the analysis of the fragments

Text

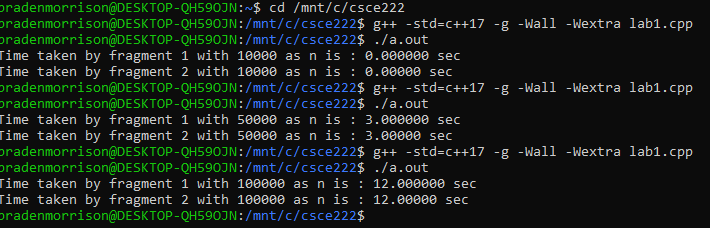
Description automatically generated

Fragment 2:

(a). O(n2)

(b). \*attached to submission on canvas

(c). As expected, the runtime for both fragments grew considerably as the input size increased. When the input size was doubled, the runtime was quadrupled, this is congruent with the analysis of the fragments



Problem 3

value == 0

for (int i = 0; i < n; i++) {

value = value \* x + a[i];

}

Problem 4

(1). O(n)

(2). The algorithm in problem 4 is faster than the algorithm I made for problem 3.