

1.
 - a. (1) Program **Foo** has a faster run time compared to Program **Bar**. For example, for an input of $N = 11,111$, Program **Foo** evaluates to 33001602.5491166 while Program **Bar** evaluates to 246908642.
 - b. (2) Program **Bar** has a faster run time compared to Program **Foo**. For example, for an input of $N = 88$, Program **Foo** evaluates to 125623.02611925. Program **Bar** evaluates to 15488.
 - c. (3) No it is not possible for **Bar** to run faster than **Foo** on all possible inputs
2. Fragment 1
 - a. (1)
 - i. $\text{sum} = 0$: $O(1)$
 - ii. $\text{int } i = 0$: $O(1)$
 - iii. $i < n, i++, \text{int } j = 0$: $O(1) * n = O(n)$
 - iv. $j < n - i, j++, \text{sum} = \text{sum} + i + j$: $(n-i)*(n)*O(1)$
 - v. $O(1) + O(1) * n + O(1) * (n-i) * n = O(n^2)$
 - b. (2) Actual run time for $n = 100$: 1192487 microseconds. Actual run time for $n = 200$: 1437129 microseconds. Actual run time for $n = 57$: 6597626 microseconds.

The screenshot shows a C++ IDE with the following code in `lab1.cpp`:

```

1  #include <iostream>
2
3
4  #include <chrono>
5  using namespace std::chrono;
6  int Fragment1()
7  {
8      int sum = 0;
9      int n;
10     std::cout << "Enter n for Fragment1: ";
11     std::cin >> n;
12     for (int i = 0; i < n; i++) {
13         for (int j = 0; j < n-i; j++) {
14             sum = sum + i + j;
15         }
16     }
17     return sum;
18 }
19
20
21
22 int main()
23 {
24     auto start = high_resolution_clock::now();
25     Fragment1();
26     auto stop = high_resolution_clock::now();
27     auto duration = duration_cast<microseconds>(stop - start);
28     std::cout << "Actual Running Time for n: " << duration.count() << std::endl;
29 }
30
31

```

The terminal output shows the execution of the program for different values of n :

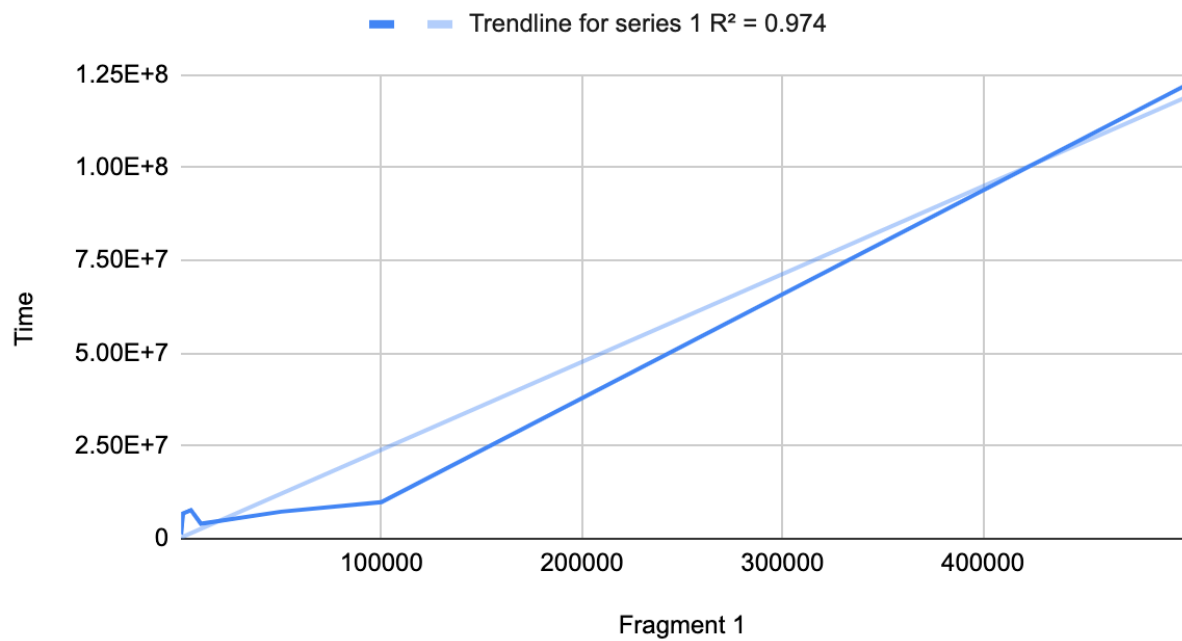
```

6 (base) david-tylerighedosa@client-10-228-25-222 Desktop % ./lab1
Enter n: 100
Actual Running Time for n: 1863946
333300
(base) david-tylerighedosa@client-10-228-25-222 Desktop % ./lab1
Enter n: 100
Actual Running Time for n: 2265726
(base) david-tylerighedosa@client-10-228-25-222 Desktop % ./lab1
Enter n: 1
Actual Running Time for n: 1028110
(base) david-tylerighedosa@client-10-228-25-222 Desktop % ./lab1
Enter n: 2
Actual Running Time for n: 1440551

```

c. (3)

Time vs. Fragment 1



Fragment 2

a. (1)

- i. $\text{sum} = 0$: $O(1)$
- ii. $\text{int } i = 0$: $O(1)$
- iii. $i < n, i++, \text{int } j = 0$: $O(1) * n = O(n)$
- iv. $j < 2 * i, j++, \text{sum} = \text{sum} + i + j$: $(2 * n) * (n) * O(1)$
- v. $O(1) + O(n) + O(1) * 2n * n = 2O(n^2)$

b. (2) Actual run time for $n = 100$: 3416351 microseconds. Actual Running Time for $n = 200$: 1534315

```

1  #include <iostream>
2
3  #include <chrono>
4  using namespace std::chrono;
5
6  int Fragment2()
7  {
8      int sum = 0;
9      int n;
10     std::cout << "Enter n for Fragment2: ";
11     std::cin >> n;
12     for (int i = 0; i < n; i++) {
13         for (int j = 0; j < 2 * i; j++) {
14             sum = sum + i + j;
15         }
16     }
17     return sum;
18 }
19
20 int main()
21 {
22
23     auto start2 = high_resolution_clock::now();
24     Fragment2();
25     auto stop2 = high_resolution_clock::now();
26     auto duration2 = duration_cast<microseconds>(stop2 - start2);
27     std::cout << "Actual Running Time for n: " << duration2.count() << std::endl;
28
29 }

```

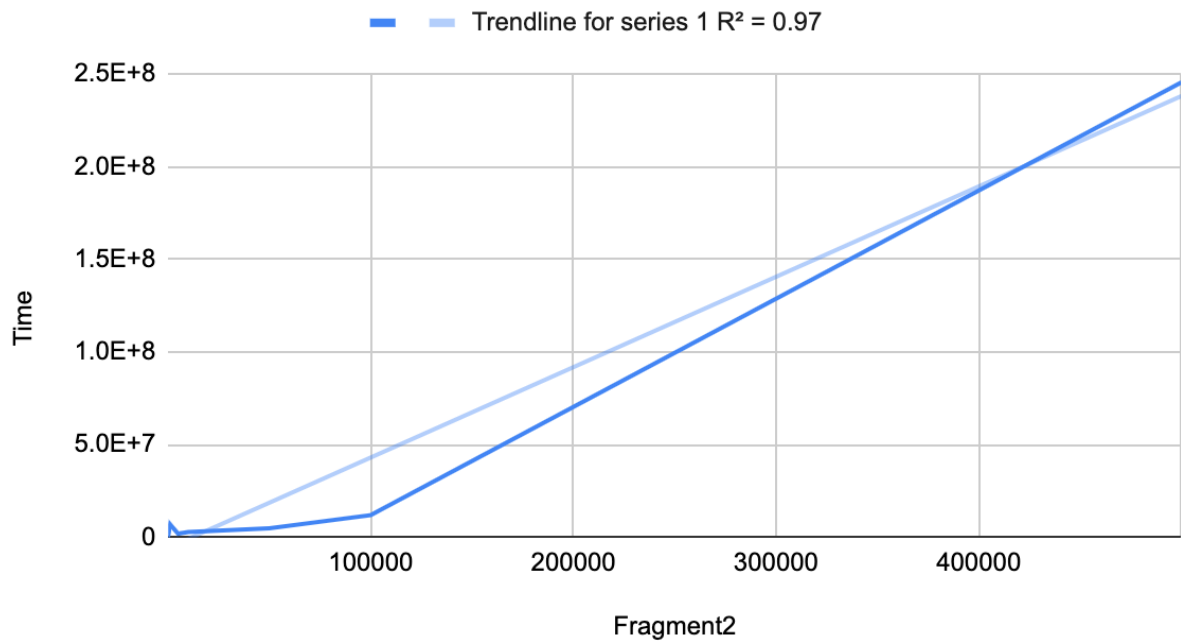
```

(base) david-tylerighedosa@client-10-228-25-222 Desktop % ./Frag2
Enter n for Fragment2: 50000
Actual Running Time for n: 4911093
(base) david-tylerighedosa@client-10-228-25-222 Desktop % ./Frag2
Enter n for Fragment2: 100000
Actual Running Time for n: 11963778
(base) david-tylerighedosa@client-10-228-25-222 Desktop % ./Frag2
Enter n for Fragment2: 500000
^Z
zsh: suspended ./Frag2
(base) david-tylerighedosa@client-10-228-25-222 Desktop % ./Frag2
Enter n for Fragment2: 500000
Actual Running Time for n: 245565926
(base) david-tylerighedosa@client-10-228-25-222 Desktop %

```

i.
c. (3)

Time vs. Fragment2



Both Fragments have the same run time or time complexity of $O(n^2)$
 3. $O(1) + O(1) + O(1) + O(1) + O(1) + O(n) * O(1) + O(1) = O(n)$ run time

```
p3.cpp
lab1.cpp 3 Frag2.cpp p3.cpp x linked_list.cpp
Users > david-tylerighedosa > Desktop > p3.cpp > main()
1  #include <iostream>
2
3  int main()
4  {
5      int value = 0;
6      int polynomial[] = {1, 2, 3, 4};
7      int n = 4;
8      int x = 5;
9      for(int i = 0; i < n; i++)
10     {
11         value = value*x + polynomial[i];
12     }
13     std::cout << value;
14 }
```

- 4.(1) value = 0: $O(1)$ + int i = n $O(1)$ + i ≥ 0 , i--: $O(n)*O(1)$ + value = value * x + a[i]: $O(1)$ = $O(n)$.
- (2) Both have the same time complexity or run time of $O(n)$ so they run the same.