Names: ____________________

The maximum contiguous subsequence sum (mcss) of an array is the largest sum you can acquire by adding up consecutive elements of an array. For example, consider the following array:

```
[5, -8, 7, 7, -1, 9, -6, -4, 5, -7]
```

The mcss of this array is 22, corresponding to the subsequence `[7, 7, -1, 9]`. Note that if the array consists of non-negative numbers then the subsequence is simply the entire array. We say that the mcss of an array consisting of all negative numbers is zero.

For this lab, you won't be writing code to solve this problem. Instead, you'll be analyzing the complexity of various solutions to this problem. Download the source file in question (linked off the course schedule):

```
.../csc207/16fa/files/MaxContiguousSubsequenceSum.java
```

The three functions—`compute1(arr)`, `compute2(arr)`, and `compute3(arr)`—all return the mcss of the given array.

In this lab, we'll measure the time complexity of these functions. Rather than using wall clock time, though, we'll measure the number of array accesses that each function makes. Let's define an array access as any case where the function reads an array value (*e.g.*, `int x = arr[`j]) or writes an array value (e.g., `arr[j] = 5`). Assuming that the array accesses dominates the runtime of the functions, then counting array accesses should be tantamount to measuring the time each function takes.

First, inspect each of the three functions and give a mathematical model (*i.e.*, a mathematical function) for the number of array accesses that each performs:

- `compute1`: n(n+1)(n+2)/6
- `compute2`: n(n+1)/2
- `compute3`: n

Please augment the three functions so that rather than returning the mcss of the given array, they report the number of array accesses each function makes. Then use these functions to collect the number of array accesses required for each function for the following array sizes: 10, 50, 100, 500, 1000, 5000, 10000, 100000.

Finally, for each function, graph the data you collected. The x-axis of your graph should be the size of the array and the y-axis should be the number of array accesses. Please include your data and your graphs on the other side of this page. You do not have to include your modified code.

(Please give your collected data and graphs below.)