**CS3353: Data Structures and Algorithm Analysis I**

**Fall 2023**

**Homework #1**

* Full name only: April Duff
* Release date: Aug 23, 2023 (Wednesday)
* Due date: **Sep 6, 2023 (Wednesday), 4:00 PM**
* It should be done INDIVIDUALLY; Show ALL your work; Submit WORD document through Canvas.
* Total: 20 pts

1. Given the following program, calculate its asymptotic complexity (Big-O) in terms of n. *Show all your work step-by-step as the examples provided in the class. Answer only without process receives half points.*

[3 pts]

public static int function(int[ ] arr) {

int n = arr.length, total = 0;

for (int j=0; j < n; j++)

total += arr[j];

return total;

}

1. FOR THE INITIALIZATION OF THE FOR LOOP
   1. How many statements will be executed?
      1. One assignment statement: j=0
   2. How many times will they be executed?
      1. Once
   3. Total number of executed assignments: 1
2. How many times will the for loop be executed?
   1. n times
3. FOR EACH ITERATION OF THE FOR LOOP
   1. How many statements will be executed?
      1. Two assignment statements: j++ and total += arr[j]
   2. How many times will they be executed over n iterations?
      1. 2\*n = 2n
4. Total number of executed assignments during the execution of the for loop:
   1. 1 + 2n 🡪 O(n)

2. Given the following program, calculate its asymptotic complexity (Big-O) in terms of n. *Show all your work step-by-step as the examples provided in the class. Answer only without process receives half points.*

[3 pts]

public static int function(int[ ] arr) {

int n = arr.length, total = 0;

for (int j=0; j < n; j += 2)

total += arr[j];

return total;

}

1. FOR THE INITIALIZATION OF THE FOR LOOP
   1. How many statements will be executed?
      1. One assignment statement: j=0
   2. How many times will they be executed?
      1. Once
   3. Total number of executed assignments: 1
2. How many times will the for loop be executed?
   1. n times
3. FOR EACH ITERATION OF THE FOR LOOP
   1. How many statements will be executed?
      1. Two assignment statements: j += 2 and total += arr[i]
   2. How many times will they be executed over n iterations?
      1. 2\*n = 2n
4. Total number of executed assignments during the execution of the for loop:
   1. 1 + 2n 🡪 O(n)

3. Given the following program, calculate its asymptotic complexity (Big-O) in terms of n. *Show all your work step-by-step as the examples provided in the class. Answer only without process receives half points.*

[4 pts]

public static int function(int[ ] arr) {

int n = arr.length, total = 0;

for (int j=0; j < n; j++)

for (int k=0; k <= j; k++)

total += arr[j];

return total;

}

1. FOR THE INITIALIZATION OF THE OUTER FOR LOOP
   1. How many statements will be executed?
      1. One assignment statement: j=0
   2. How many times will they be executed?
      1. Once
   3. Total number of statements: 1
2. How many times will the outer for loop be executed?
   1. n times
3. FOR EACH ITERATION OF THE OUTER FOR LOOP
   1. How many statements will be executed?
      1. One assignment statement: j++
   2. How many times will it be executed over n iterations?
      1. 1\*n = n
4. FOR THE INITIALIZATION OF THE INNER FOR LOOP
   1. How many statements will be executed?
      1. One assignment statement: k=0
   2. How many times will they be executed?
      1. n times
   3. total number of statements: 1\*n = n
5. FOR EACH ITERATION OF THE INNER FOR LOOP
   1. How many statements will be executed?
      1. Two assignment statements: k++ and total += arr[i]
   2. How many times will they be executed?
      1. Depending on the value of j, where k <= j
   3. Total number of statements: 2 \* (1 + 2 + 3 + … + n-1) = n(n-1)
6. Total number of executed assignments:
   1. 1 + n + n + n(n-1) = 1 + 2n + n2 – n = n2 + n + 1 = O(n2)

4. Given the following program, calculate its asymptotic complexity (Big-O) in terms of n. *Show all your work step-by-step as the examples provided in the class. Answer only without process receives half points.*

[4 pts]

public static int function(int[ ] arr) {

int n = arr.length, prefix = 0, total = 0;

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

prefix += arr[j];

total += prefix;

}

return total;

}

1. FOR THE INITIALIZATION OF THE FOR LOOP
   1. How many statements will be executed?
      1. One assignment statement: j=0
   2. How many times will they be executed?
      1. Once
   3. Total number of executed assignments: 1
2. How many times will the for loop be executed?
   1. n times
3. FOR EACH ITERATION OF THE FOR LOOP
   1. How many statements will be executed?
      1. Three assignment statements: j++, prefix += arr[j], and total += prefix
   2. How many times will they be executed over n iterations?
      1. 3\*n = 3n
4. Total number of executed assignments during the execution of the for loop:
   1. 1 + 3n 🡪 O(n)

5. Given the following program, calculate its asymptotic complexity (Big-O) in terms of n. *Show all your work step-by-step as the examples provided in the class. Answer only without process receives half points.*

[4 pts]

public static int function(int[ ] first, int[ ] second) {

int n = first.length, count = 0;

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

int total = 0;

for (int j=0; j < n; j++)

for (int k=0; k <= j; k++)

total += first[k];

if (second[i] == total)

count++;

}

return count;

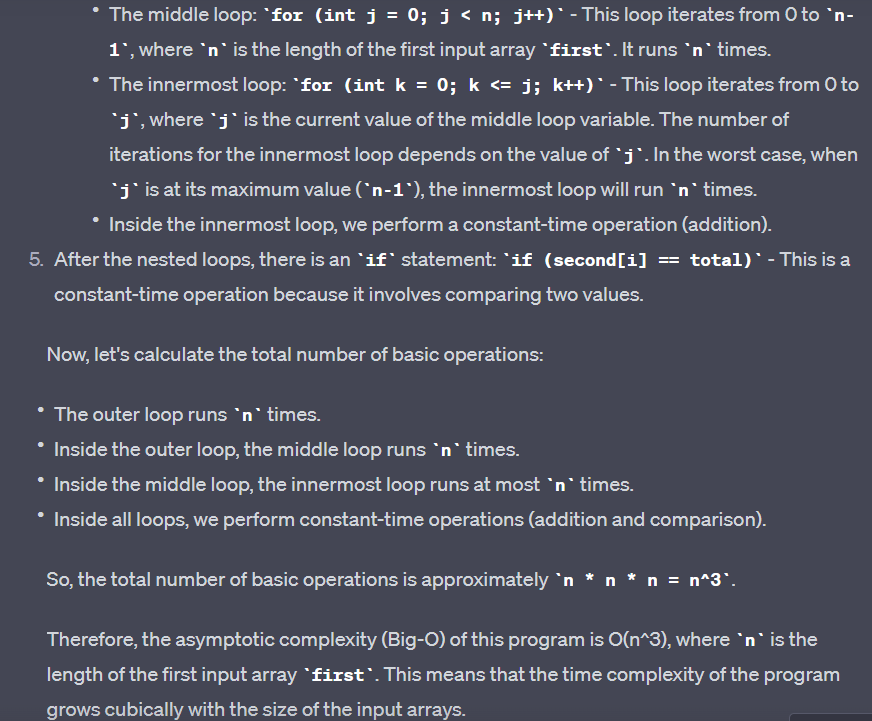
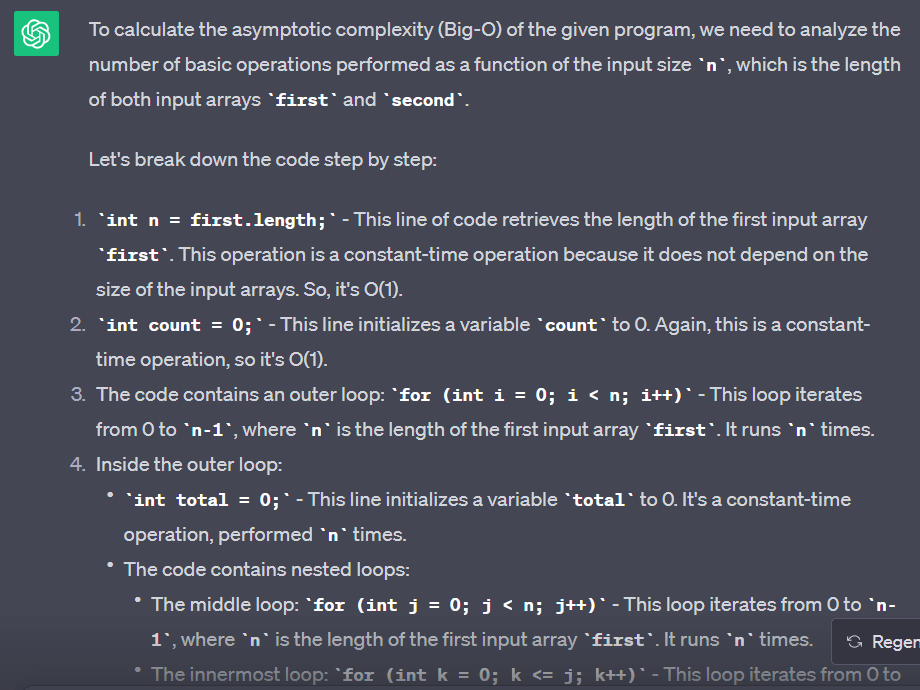
}

1. FOR THE INITIALIZATION OF THE OUTER FOR LOOP
   1. How many statements will be executed?
      1. Two assignment statements: i=0 and total = 0
   2. How many times will they be executed?
      1. i=0 will execute once and total=0 will execute n times
   3. Total number of statements: 2
2. How many times will the outer for loop be executed?
   1. n times
3. FOR EACH ITERATION OF THE OUTER FOR LOOP
   1. How many statements will be executed?
      1. One assignment statement: i++
   2. How many times will it be executed over n iterations?
      1. 1\*n = n
4. FOR THE INITIALIZATION OF THE MIDDLE FOR LOOP
   1. How many statements will be executed?
      1. One assignment statement: j=0
   2. How many times will they be executed?
      1. Once
   3. Total number of statements: 1
5. How many times will the middle for loop be executed?
   1. n times
6. FOR EACH ITERATION OF THE MIDDLE FOR LOOP
   1. How many statements will be executed?
      1. One assignment statement: j++
   2. How many times will it be executed over n iterations?
      1. n\*n = n2
7. FOR THE INITIALIZATION OF THE INNER FOR LOOP
   1. How many statements will be executed?
      1. One assignment statement: k=0
   2. How many times will they be executed?
      1. n times
   3. total number of statements: 1 \* n = n
8. FOR EACH ITERATION OF THE INNER FOR LOOP
   1. How many statements will be executed?
      1. Two assignment statements: k++ and total += first[k]
   2. How many times will they be executed?
      1. Depending on the value of j, where k <= j
   3. Total number of statements: 2 \* (1 + 2 + 3 + … + n-1) = n(n-2)(n-1)
9. Total number of executed assignments:
   1. 2 + n + n + n2 + n + n(n-2)(n-1) = 2 +3n + n2 + n3 – 3n2 + 2n = n3 - 2n2 + 5n + 2 = O(n3)

6. Use an AI tool, such as ChatGPT, Bing AI, or Bard, to generate a “solution” for the program in Question 5*.*

*6.a. Attach the screenshot of “solution” generated by AI tool.*

[1 pt]



*6.b. Compare your solution with the “solution” generated by AI tool, and summarize your findings* (at least 50

words).

[1 pt]

When comparing my solution to ChatGPT’s solution, I first noticed the similarities between the analysis processes. Both take into account the initialization and iteration of each for loop. Another thing I noticed was it considered the greatest number of times the inner loop could be performed whereas my solution depended on the value of i. For the final calculation of all executed assignments, ChatGPT multiplied how many times each loop ran inside of each other; so, it was n \* n \* n.