**Activities**

**1.a)** the output statement is executed ‘ n \* n = n^2’ times and has a time of **O(n^2).**

**b)**the output statement is executed ‘ n \* 2 = 2n’ times. This algorithm has a time of **O(n).**

**c)** Starts from ‘n – 1’ and iterates until ‘I’, which decreases by 1 in each iteration of the outer loop (‘ for (int I = 0; I < n; i++)’). To find the total number of iterations, we can sum the iterations of the inner loop for ‘I’ from 0 to ‘n – 1 ‘and turns out to be ‘(n + 1) \* n / 2’ which is proportional to O(n^2).

**d)** The total number of iterations for all values of ‘I’ from 1 to ‘n – 1’ can be calculated as the sum of the first ‘ n -1 ‘ natural numbers , which is proportional to O(n^2).

**2.a)** the elements of ‘anArray’ will be shifted one position to the right starting from index 3.but, a problem in the loop is it will throw an ‘**ArrayIndexOutOfBoundsException**’ because it tries to access an index that is out of bounds.Specifically ‘I’ becomes equal to ‘anArray.length – 2’ , ‘i+ 1’ will become equal to ‘anArray.length – 1’, which is out of bounds for the array.So, the loop will not execute properly, and the ‘anArray’ will remain unchanged:   
{0,1,2,3,4,5,6,7}

**b)** the same as before. The element at index ‘I’, moving from right to left. However since the loop stops at index 4 (‘I > 3’), the element at index 3 remains unchanged. So, the resulting ‘anArray’ will be :{0,1,2,3,4,5,6}. So after the execution of the second loop, the elements of ‘anArray’ will be ‘{0,1,2,3,4,5,6}’. The first loop doesn’t affect the array.

**3.a)** …There are 2 operations inside and performed 2 times and it time complexity is O(n)+2n

**b)**…There are 5 operations(2accesses, 2 multiplications, and 1 addition) and the time complexity is O(r1 x c1 x c2)

**c)** O(n)+ 3n  
**d)** O(log2(n))

**4.**

* **O(1)- Constant Time Complexity: O(1)**
* **O(log n) – Logarithmic Time Complexity : O(log n)**
* **O(n)- Linear Time Complexity: O(n)**
* **O(n log n) – linearithmic Time Complexity: O(n log n)**
* **O(n^2)- Quadritic Time Complexity; O(n^2)**
* **O(n^3)- Cubic Time Complexity: O(n^3)**
* **O(2^n)- Exponential Time Complexity: O(2^n)**

**5.** An Abstract Data Type is a mathematical model that defines a set of operations on data structure without specifying how these operations are implemented.  
example:

**A screen shot of a computer program

Description automatically generated**

|  |  |  |
| --- | --- | --- |
| Features | List | ArrayList |
| Interface/Class | Interface | Class |
| Implementation | Can be implemented by various classes | Implemented using an array internally |
| Resizable | yes | Yes |
| Memory | Dynamically allocated, no contiguous in memory | Contiguous memory allocation |
| Random Access | Yes(via index) | Yes(Faster to array-backed structure) |
| Insertion/Deletion | Can be slower due to shifting elements | Can be slower for iniserting/ deleting in middle of the list |
| Memory Efficiency | Consumes more memory due to additional pointers | More memory=efficient due to contiguous allocation |
| Suitable for | Use when flexibility or implementation independence is desired | Use when random access, fast iteration, and less memory consumption are priorites |

6.

7. A computer screen shot of a program

Description automatically generated