CS401

(Lab_3)

PART-1

METHOD A:

The nested loops iterates 'n' times each, resulting in a total of n * n iterations. Therefore, the time complexity of methodA is O(n^2).

METHOD B:

The nested loops iterates 'n' times each, resulting in a total of n * n * n iterations. Therefore, the time complexity of methodB is O(n^3).

METHOD C:

The nested loops consist of a constant factor loop (for i = 0; i < 100 * n) and one loop with a decreasing geometric progression (j /= 2). Therefore, the time complexity of methodC is O(n * log(n)).

To Summon up:

Method A – O(n^2)
Method B - O(n^3)
Method C - O(n * log(n))

PART-2

The time complexity of the provided selection sort algorithm is O(n^2), where n is the number of elements in the array.

Explanation:

Outer loop: Runs for (n-1) iterations. This loop represents the current position.

Inner loop: (minIndex method) runs for (n-1) + (n-2) + (n-3) + ... + 1 iterations in the worst case. This loop finds the index of the minimum element in the unsorted array.

In each iteration of the outer loop, a swap operation is performed, which takes constant time.

So, the overall time complexity is determined by the nested loops, resulting in O(n^2). The space complexity of the algorithm is O(1) as it uses a constant amount of additional space for variables regardless of the input size.