DSA HOMEWORK 2 QUESTION 3 – ORDER OF GROWTH TIME AND COMPLEXITY

THE WORST CASE BIG 0 RUNTIME OF CODE SNIPPETS (A) – (E)

a) The first for loop will run for "n" instances in which n is the period/length of the listing. When "i" may be zero the primary nested for loop will run "n" instances. The subsidiary nested for loop will run for "n/2" instances. The overall time complexity is n((n+n)/2) that is identical to (n^2) +(n^2)/2. So, the Big O runtime is equivalent to O(n^2).

b) Every time the loop runs the "n" receives division via way of means of 2. For "n", i = 0. , For "n/2", i = 1. When "n/4", i = 2. Hence for each increment of i, n gets divided by a multiplicand of 2. Time complexity is 2n. Big O notation is equivalent to O(n).

c) The first for loop runs for "n" instances in which n is the period/length of the listing/array. The nested for loop runs for "n" instances. Time complexity for binary search is log(n). The subsidiary nested loop has selection sort which takes n^2 runtime. So, time complexity is equivalent to n^2log(n) + n^3. Big O notation is credited as O(n^3).

d) As for loop is strolling for n^2 time due to nested listing or square listing, the time complexity is equivalent to "n^2". The merge type has time complexity "nlogn". So, overall time complexity is n^2 + nlog(n). Big O notation is equivalent to O(n^2).

e) In this code, the counter is getting extended via way of means of a multiplicand of 2 so the time complexity for the while loop is "log n". The time complexity for binary search is "log n". The overall time complexity and Big O notation is equivalent to O((logn)^2).