1. L[n-1] is A O(1)

This is because the operation is obtaining the last item in L, so the time is constant as long as we know N. The only time it would be an issue is when N is zero, which would produce and error. The operation is just retrieving data from a specified point in the list.

1 in L is C O(n)

This is because the operation will have to search through, at most, all of L to determine if there is a 1 inside of L. And all of L requires and N amount of searches.

1. T(n)= n3+20n+1 is not O(n2 ), because:

N3+20n+1<= cn2

/n2

N + (20/n) + (1/n2) <=c

The above expression would require c to change, but it is supposed to be a constant, so O(n2) is not the O notation.

1. T(n)= n3+20n is Ω(n2 ), because:

Since every n is going to be positive, and n3 is also n2\*n. n2 is a lower or equal bound(big-Omega) since n2\*n>=n2 for all positive values of n.

1. 264-1

21000

Log n

Log (n2)

Log(n3)

N

2log(n)

1000n

N log n

0.0001n2

N2logn

N3

N3logn

N100

(3/2)n

2n

N2n

4n

nn

1. Big-O is a representation of the growth rate of a function’s worst-case scenarios as the function size that is operated on is increased.

5n+10 <= c\*n2

­/n2

5/n+10/n2 <=c

N 🡪 Infinity

0+0<=c

C=15, since (1) gives 15 as a value. N0 is 1.

There will be a positive value of c that is greater than 0.

Since both c and n­­0 exist, O(n2) exists for 5n+10.

1. Code 1: O(n), because the code uses each element in S only once.

Code 2: n/2, which “reduces” to O(n) since it is only adding about half of the elements in n. This division does not affect the big O notation for growth though.

Code 3: The big O is O(n2), this is because the nested for loop’s change with n. The limiters of both loops are based on n, so the loop is a variant of n\*n in time.

Code 4: O(n), since the code goes through n only once and saves data to multiple variables, unlike the code before that used a second for loop.

Code 5: O(n3), this is because of the three nested for loops. The two loops going up to n, definitely give it at least a O(n2), but then the other loop for the worst case scenario will provide a O(n3) time complexity.