1. a. O(n^2)

b. O(n)

c. O(n^2)

d. O(n)

1. {1,2,3,4,2,2,2}

{1,2,3,4,1,1,1}

1. a. T(n) = 2n + 2

O(n)

b. T(n) = 3\*n1\*n2\*n3 + 2

O(n1\*n2\*n3)

c. T(n) = 1 + 3n

O(n)

d. T(n) = 1 or infinite depending on whether the number is positive or negative

O(1) or O(infinite)

1. O(1) insert into queue

O(log n) binary search

O(n) linear search

O(n^2) quick sort worst case

O(n!) emulating all partitions of a set

1. Abstract data types a way of classifying data structures based on how they are used and the behaviors they provide

Implementation: stacks, queue, linked list, graphs, hash tables

1. List is a child interface of Collection. It is an ordered collection of objects in which duplicate values can be stored. Since List preserves the insertion order, it allows positional access and insertion of elements. List interface is implemented by the classes of ArrayList, LinkedList, Vector, and Stack. List is an interface, and the instances of List can be created by implementing various classes.

