16.  Three algorithms A, B, and C, are under consideration for the Insert operation of a particular data set.

(a)  Calculate the value of each algorthm when *n*, the number of nodes in the data structure, is equal to 1,000,000.

1. Algorithm A: 23n + 36n2 3.6X1013 + 2.3X107 = 3.6X1013
2. Algorithm B: 6 + nlog2(n) + n 6 + 2X107  + 1X106  = 2.1X107
3. Algorithm C: log2n + 36n2 20 + 36X1012 = 3.6X1013

(b)  What is the Big-O value for each algorithm.

1. Algorithm A: 23n + 36n2 O(36 n2)
2. Algorithm B: 6 + nlog2(n) + n O(nlog2(n)) *// nlogn is faster than n but slower than n2 for large n.*
3. Algorithm C: log2n + 36n2 O(36 n2)

17.  A 1000 element array is used to store integers in ascending order. The array is to be searched using the binary search algorithm for the two integers 5215 and 7282. How many elements of the array would be examined by the algorithm to locate? *For both of these, convert the address to binary. The result will tell us where it is in the array. Assuming the resulting array us full. (I would set it up as having 1024 elements, probably, since that's a power of two.)*

(a)  the integer 5215 stored in element 499? *Seven passes. Assuming the array is full.*

(b)  the integer 7282 stored in element 686? *Six passes.*

18.  What is the maximum and minimum number of times the search loop will execute when searching through an array of 1,048,576 = 220 integers if the search algorithm is

(a)  the binary search? *minimum =1, maximum = 20. (Assuming the array is full). The maximum could be less if the array has many empty nodes.*

(b)  the sequential search? *minimum =1, maximum = 1048576. (Again smaller if the array is partially empty and we can recognize an empty slot without examining its contents. This is something we can be confident of if "Garbage Collection" was done on delete operations.)*

20.  Observations of the “traffic” on a data structure over a certain period of time indicate that 500 Insert operations, 500 Delete operations, 700 Fetch operations, and 200 Update operations were performed on a data set. If Insert operations take 10 nanoseconds, Delete operations take 250 nanoseconds, Fetch operations 200 nanoseconds, and Update operations take 300 nanoseconds, determine:

(a)  the probability of performing a Fetch operation over the observation period.

*The probability of a Fetch operation is 0.42 = 42%.*

(b)  the average speed, in nanoseconds, of the data structure over the observation period.

*The average time per operation is 0.173 μsec.*

21.  Calculate the density of a data structure whose data set consists of 1,000,000 nodes, assuming the structure requires 1,000,000 bytes of overhead to maintain itself, and:

(a)  each node in the data set contains 2000 information bytes.

*The density is 0.9995.*

(b)  each node in the data set contains 20 information bytes.

*The density is 0.952. This is hardly a surprise since there are so many fewer data bytes as compared with the overhead. We would expect a smaller density.*

23.  State the Java code to declare an array of 100 integers named ages.

int[] ages = new int[100];

24.  State the Java code to declare an array of three Listing objects named data that are initialized with the no-parameter constructor.

Listing data = new Listing[3];

for(int i=0;i<3;i++) {data[i] = new Listing(); }

26.  Two objects, objectA and objectB, are objects in the class Listing. The object objectA is copied to objectB. How many objects exist after the copy, if the copy is performed as:

(a)  a deep copy?

*A deep copy would leave us with two copies of the same object in different memory locations.*

(b)  a shallow copy?

*A shallow copy will leave us with one copy of the object and (potentially) an object with no pointer. Sometimes this may be desirable.*

27.  Of the two types of copies discussed in the previous exercise, which one produces a clone (an exact duplicate of an existing object)?

*The shallow copy only produces a new link to the data. It is the deep copy which actually reproduces the data.*

28.  Give only the signature of a method named *deepCopy*, that clones an object in the class Listing sent to it as a parameter and returns a reference to the clone. Do not provide code. Only the signature!

Listing deepCopy(Listing x) {}

29.  Give the Java invocation to clone the object objectA (using the method discussed in the previous exercise) and store a reference to the clone in the variable: newListing.

Listing newListing = deepCopy(objectA);