3.3) 2, log3n, log2n, n2/3, 20n, 4n2, 3n, n!

3.8)

a) The upper Bound c = c1

The lower Bound c=c1

b) The upper bound is c = c2 +c3

The lower bound is c = c2

c) The upper bound is (O nlogn) for no > a and c = c4 + c5 , where a is base of the logarithm

The lower bound is big Theta(nlogn) for no > a and c = c4 where a is the base of the logarithm

4.1) Given the list

if(arr.contains(n)) { // where n is the number thing you’re looking for

arr.remove(indexOf(n))

}

return arr;

}

}

4.1)

1. Array Based = 16 Linked list = 12 160/12 = 1.3333 so when n >13.33333
2. Array Based = 60 Linked List = 6 60/6 = 10 so when n > 10
3. Array Based = 30 Linked list = 5 30/5 = 6 so when n > 6
4. Array based = 1280 Linked list = 36 1280/36 = so when n > 35.55555

\*\* The array-based implementation is more space efficient in any particular situation pass n \*\*