Big O Notation Assignment

Pre-Homework

1. Write a function that take two numbers, k and l, as parameters and returns the result of: .

(In Java)

public double Res(double k, double l){

double t = ((Math.pow(4,l)) + k);

double b = ( 3 \* (Math.pow(k,3))) + l;

double x = t/b;

return x;

}

1. Write a function called *mean*  that returns the mean value of a sequence of numbers.

(In Java)

public int Mean(int[] A){

int total = 0;

for (int element: A){

total += element;

}

return total;

}

Actual Homework

1. Describe the runtime bounds of the linear search algorithm.

BOOL LINEAR-SEARCH ( ITEM, LIST[ ])

( n ) for i ← 0 to Length[ LIST[] ]

( n ) if ( LIST[ i ] = ITEM)

( 1 ) return true

return false

Time-complexity: O(n) .

1. Describe the runtime bounds of the duplicate finder algorithm.

BOOLEAN EXAMINEFORDUPLICATES (LIST[])

(n) for i ← 1 to Length[LIST]

(nn) for j ← i+1 to Length[LIST]

(nn) if LIST[i] = LIST[j]

(1) return true

return false

Time-complexity: O(n2) .

1. Additional homework:

Write a function that determines the critical value at which the relative runtime of two linear algorithms swap.

INT DetCritVal ( M1, K1, M2, K2 )

Switch ← False

N ← 0

If (((M1\*N) + K1) > ((M2\*N) + K2))

While (!Switch)

If (((M1\*N) + K1) < ((M2\*N) + K2))

Switch ← True

else

N++

else

While (!Switch)

If (((M1\*N) + K1) > ((M2\*N) + K2))

Switch ← True

else

N++

return N