**Name: - Milkias Ghebremariam**

**Student Id: -109364**

**Name: -Tomas Melake**

**Student Id: -109334**

Solution to W1D3

Solution1,

(a)

Algorithm removeDups4(A,n)

Input : An array A with n>0 integers

Output:An array with all duplicates in A removed

hs🡨new HashSet()

length🡨n-1

for i 🡨0 to length do

hs.add(A[i])

b🡨enw Array(hs.size())

re🡨new ArrayList(hs)

for i🡨0 to re.size do

b[i]<-re.get(i)

return b

(b)

Time complexity is O(n)

(c)

time complexity of Algorithm #4 is O(n) same as algorithm #3.

Solutin 2.

Algorithm beautifulMul(A,n)

Input : An array A with n integers

Output: sum of n Integers

mul🡨0

length🡨n-1

for i 🡨0 to length do

mul🡨mul\*A[i]

return mul

This algorithm has a time complexity of O(n) for the worst-case best case and average case. This is because the algorithm has to traverse the whole elements to multiply them.

Solutin 3.

The Order of time complexity is 2^n ,2^n+1 is O(2^n) then 2^2n is O(4^n) then 2^2^n is O(2^2^n) in increasing Order respectively.

Soltutin 4.

O(1),O(nlog),O(nlogn),O(n^2), O(n^3), O(2^n)

Example:O(nlogn):Quicksort,megeSort

O(logn):

O(1):Add method of List, insertion & deletion of HashTable and doubly linked list.

O(n):doubley linked list search,

O(n^2):Smooth sort,Bubble,SelectionSort,insertion sort.

O(n^3): Naive multiplication of two *n*×*n* matrices.

O(2^n):recursive Fibonacci(n)

Solution 5:

(a)we cant use masters theorem for Fibonacci .

T(n) of recursive Fibonacci algorithm is O(2^n)

Tight upper bound is T(n)=O(1.6180)^n

(b) T(1)=d;

T(n)=c+T(n/2)

a=1, b=2, k=0, b^k=1

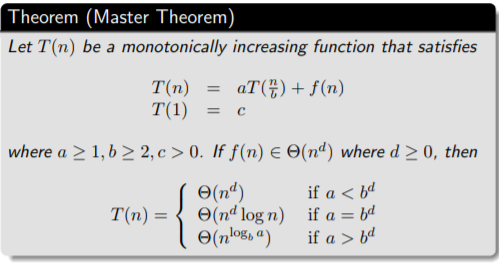
a=b^k=1

T(n)=ϴ( n^k log n)

T(n)= ϴ( log n )

Solution 6:

Master theorem



Emaples

Example 1,

Let T(n) = T n 2 + 1 2 n 2 + n. What are the parameters?

a = 1 b = 2 d = 2

Therefore which condition?

Since 1 < 2^2 ,

case 1 applies. Thus we conclude that

T(n) ∈ Θ(n^d ) = Θ(n^2)

Eaxmple 2.

Let T(n) = 2T n 4 + √ n + 42.

What are the parameters?

a = 2 ,b = 4, d = 1/2

Therefore which condition?

Since 2 = 4^(1/2) ,

case 2 applies.

Thus we conclude that

T(n) ∈ Θ(n^d log n) = Θ(√ n log n)

Eaxmple 3.

Let T(n) = 3T n 2 + 3 4 n + 1.

What are the parameters?

a = 3, b = 2 ,d = 1

Therefore which condition?

Since 3 > 2^1 ,

case 3 applies. Thus we conclude that

T(n) ∈ Θ(n^(logb a) ) = Θ(n log2 3 )

Note that log2 3 ≈ 1.5849 . . ..

Can we say that T(n) ∈ Θ(n^(1.5849))