Question 1:

Algorithm removeDups1(A,n)

Input: An array A with n > 0 integers

Output: An array B with all duplicates in A removed

D<- new hashSet

For i<-0 to n-1 do

If d not contains A[i]

D add A[i]

sizeHashSet<-size of hashset

B<- new Array(sizeHashSet)

For M<-0 to sizeHashSet

B[k]=d[M]

Return B

Analysis of Algorithms

Analysis

T = running time of removeDups4

O(n) initialization + single for loop of size n +initialization+ O(n) copy operation

=> T(n) is **4 * O(n) = O(n)** Therefore, the running time of Algorithm #3 is O(n)

Question 2:

Algorithm beautiful(A n)
Input : An integer array with n elements
OutPut: the smallest number
→Assume we have a method to Findsmallest input Array and Array index and Array ending index and return the smallest
n ← A.length
if <= 1
return A[0]
m1 ← Findsmallest(A, 0, n/2)
m2 ← Findsmallest(A, (n/2)+1, n)

```
If m1<m2
        return m1
else:
        return m2
```

Best case and Worst Case in this Algorithm is equal because even thou we have a sorted Array this Algorithm will not just simply take the First Number Assuming it's the smallest so. Whenever we have an input to this Array it will divide them and find the min from them and compare the min from both of them.

Question 3:

→ 2^n < 2^n+1 < 2^2n < 2^2^n

Question 4:

→O(1) Print("Hello world")

→O(n) Linear Search

→O(log n) Binary Search

→O(n log n) Merge Sort

→$O(n^2)$ Selection Sort

→$O(n^3)$

→$O(2^n)$ Recursive Fibonacci

Question 5:

A. Fibonacci f(n) shown in slide 48 can't be determined using the Master Theorem.

B. Binary Search → T(n) = c + T(n/2)

Theorem. Suppose $T(n)$ satisfies

$$T(n) = \begin{cases} d & \text{if } n = 1 \\ aT(\lceil \frac{n}{b} \rceil) + cn^k & \text{otherwise} \end{cases}$$

where k is a nonnegative integer and a, b, c, d are constants with $a > 0, b > 1, c > 0, d \geq 0$. Then

$$T(n) = \begin{cases} \Theta(n^k) & \text{if } a < b^k \\ \Theta(n^k \log n) & \text{if } a = b^k \\ \Theta(n^{\log_b a}) & \text{if } a > b^k \end{cases}$$

a=1, b=2, k=0
$a = b^k$ → $1 = 2^0$

Binary Search is then O (log n).

Question 6:

A. T(n) = 2T(n/2)+n
a = 2 b = 2 k = 1
$b^k = 2^1 = 2 = a$

Applying the Master Formula:
$O(n^1 \log n) = O(n \log n)$

B. T(n) = T(n/2)+n

a=1, b=2, k=1
$b^k = 2^1 = 2 > 1 = a$

Applying the Master Formula:

O(n)

C. T(n) = 4T(n/2)+n
a=4, b=2, k =1

$b^k = 2^1 = 2 < 4 = a$

Appling Master Formula:

$$\mathbf{O\ (n^{Log_b a})}$$

$$\mathbf{Log_b a = log_2 4 = 2}$$

$$O\ (n^2)$$