CSE 221 (ALGORITHMS)

LAB 1

Done by:

Nabil Hasan

ID: 19301222

Lab sec: 05

Theory sec: 05 (Faculty : Shaily Roy [SRO])

Ahmad Zubair

ID: 19101147

Lab sec: 05

Theory sec: 08 (Faculty : Sajed Haque[SAQ])

Submitted to:

Sabrina Zaman Ishita [SZI]

AKM Amanat Ullah

Primality Testing

Worst Case complexity for Naïve Approach:

$O(n^2)$

Worst Case complexity for Optimal sieve:

O(n* root(n))

Recursion Tree Time Complexity

1. T(n) = T(n/2)+n-1, T(1) = 0

T(n) = T(n/2)+n-1

=T(n/22) + (n/2) -1+n-1

=T(n/23)+ T(n/22) -1 + (n/2) -1+n-1

……

……

=T(n/2k)+(n/2k-1 +…. (n/22) + (n/2) +n) – (1+1+…+1)

Considering,

n/2k=1,

.'. T(n)= T(1)+ n(1/2k-1 +….1/22+ ½ +n) –(1+1…+1)

= 0+n*(1)-c

=n-c

=O(n)

Worst time Complexity= O(n)

2. T(n) = T(n-1)+n -1, T(1) = 0

T(n) = T(n-1)+n -1

=T(n-2)+(n-2)+(n-1)

=T(n-3)+(n-3)+(n-2)+(n-1)

…..

…..

T(n)=1+2+3+…..+(n-3)+(n-2)+(n-1)

=n(n+1)/2

=$O(n^2)$

Worst time Complexity= $O(n^2)$

3. T(n)=T(n/3)+2T(n/3)+n

=3T(n/3) +n

.'. $3T(n/3)= 3^2T(n/3^2)+ 3(n/3)$

$3^2T(n/3^2)= 3^3T(n/3^3)+3^2(n/3^2)$

….

….

$3^kT(n/3^k)= 3^{k+1}T(n/3^{k+1})+ 3^k(n/3^k)$

Considering,

$n/3^{k+1}=1$

.'.$k+1= \log_3 n$

Adding,

.'.$T(n)=3\log_3 n+(n+n+n+…….+n)$

$T(n)= n+ (n* \log_3 n)$

$=O(n* \log_3 n)$

Worst time Complexity=$O(n* \log_3 n)$

4. T(n)=2T(n/2) + n^2

$2T(n/2)= 2^2T(n/2^2)+2(n/2)^2$

$2^2T(n/2^2)=2^3T(n/2^3)+2^2(n/2^2)^2$

….

…..

$2^{k-1}T(n/2^{k-1})= 2^kT(n/2^k)+ 2^{k-1}(n/2^{k-1})^2$

Considering,

$n/2^k=1$

$T(n)=n^2+n^2/2 +n^2/2^2+ n^2/2^3+ n^2/2^4+…….+n^2/2^{k-1}$

$T(n)= n^2(1+1/2 +1/2^2+ 1/2^3+ 1/2^4+…….+1/2^{k-1})$

$=O(n^2)$

Worst time complexity= $O(n^2)$

Pseudocode to Coding

```
import java.util.Scanner;
public class Lab1Task3{
public static void main(String []args){
  System.out.println("Please enter the value of n:");
  Scanner bot= new Scanner(System.in);
  int n=bot.nextInt();
  int a=n;
  int sum=0;
  while (n>0){
    int r=n%10;
    sum=sum+r*r*r;
```

```
    n=n/10;
  }
  if(a==sum){
    System.out.println("Armstrong Number");
  }
  else{
    System.out.println("not an Armstrong Number");
  }
}
}
```

****p.s- separate .java file is also submitted.**