

1.

Unordered Complexities	Ordered Complexities
N	2/N
\sqrt{N}	37
$N^{1.5}$	\sqrt{N}
N^2	N
$N \log N$	$N \log(\log(N))$
$N \log(\log(N))$	$N \log N$
$N \log^2 N$	$N^{1.5}$
2/N	$N \log^2 N$
2^N	N^2
$2^{(N/2)}$	$N^2 \log N$
37	$2^{(N/2)}$
$N^2 \log N$	2^N

2. 35 seconds for $n=20$, how much time for $n=100$?

- $O(N) = \left(\frac{35}{20}\right) * 100 = 175 \text{ seconds}$
- $O(N + \log N) = 175 + \log_2(175) = 182.45 \text{ Seconds}$
- $O(N^3) = 175^3 = 5,359,375 \text{ Seconds}$
- $O(2^N) = 2^{175} = 4.789 * 10^{52}$

3. Evaluate $\sum_{i=0}^{\infty} \frac{2xi}{4^i}$

$$= 2x \sum_{i=0}^{\infty} \frac{i}{4^i}$$

$$S = \frac{1}{4} + \frac{2}{4^2} + \dots$$

$$4S = 1 + \frac{2}{4} + \frac{3}{4^2} + \dots$$

Subtract S from $4S$

$$3S = 1 + \frac{1}{4} + \frac{1}{4^2} + \frac{1}{4^3} + \dots$$

$$3S = 4/3$$

$$S = 4/9$$

$$\text{Solution} = \frac{8x}{9}$$

4. Recursive Function to calculate the height of a binary tree

```
Int findHeight(Node *root)
```

```
{
```

```
    Int depth1 = -1, depth2 = -1;
```

```

    If(root==null)
        Return 0;
    Else
    {
        Depth1 = (root->left);
        Depth2 = (root->right);

        If(Depth1>Depth2)
            Return Depth1+=1;
        Else
            Return Depth2+=1;
    }
}

```

5.


```

While(n>0){
Move(n-1,src,dst,temp)
"Move nth disk from src to dst"
n- -
src is now temp
temp is now src
}

```
6. Git is a tool that allows us to track changes in code for use in software development. We will be using it to work on and submit assignments.
7. DDD is the data display debugger and serves as a graphical interface for various command line debuggers such as GDB. Two features are the graphical interface and break points
8. Argc and argv get set when you run the program from the command line. Argc gets sets with an integer equal to the number of arguments given when running the program. Argv gets set with the arguments as strings.