DAA Tutorial 2

1. What is the time complexity of below code & how?

```
void func(int n)
{
        int j=1; i=0;
        while (i<n);
        i=i+J;
        J++;
    }
```

$J=1$ $\quad i=1$

$J=2$ $\quad i=1+2$

$J=3$ $\quad i=1+2+3$

$i=1+2+3+\dots m<n$

$\Rightarrow \frac{m(m+1)}{2}<n$

$\frac{m^2+m}{2}<n \Rightarrow m^2<\sqrt{n}$

$m \simeq \sqrt{n}$

By Summation method

$= \sum_{i=1}^{m} 1 \Rightarrow 1+1+1+\dots m = 1+1+1+\dots\sqrt{n}$

$=\sqrt{n}$

$T(n)=\sqrt{n}$

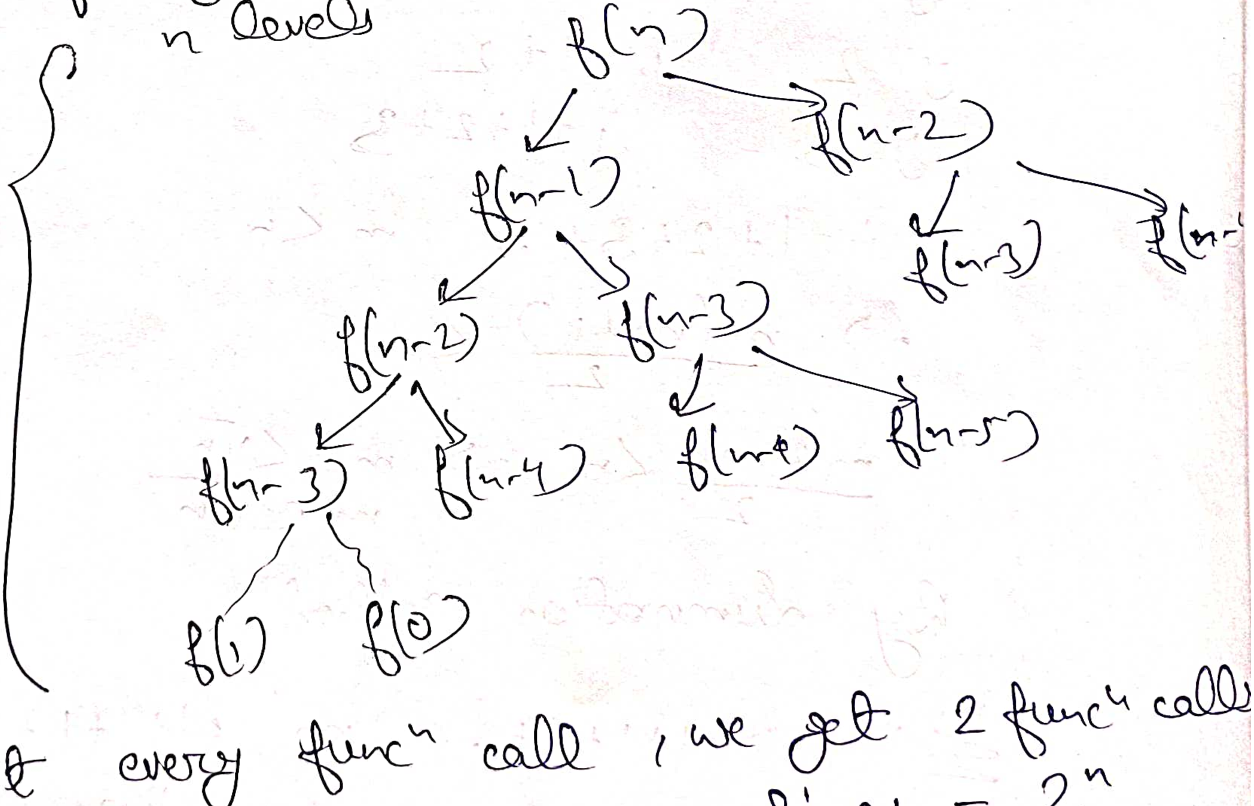
2. Write recurrence relation for the recursive func^n that prints fibonacci series. Solve the recurrence relation to get time complexity of the program. What will be the space complexity of this program & why?

for fibonacci series

$f(n) = f(n-1) + f(n-2)$

$f(0) = 0, \ f(1) = 1$

forming a tree



At every func^n call, we get 2 func^n calls

for n levels, 2×2.....n times = 2^n

$T(n) = O(2^n)$

Max. space: Space complexity depen on the max. depth of tree, so space complexity $= O(n)$.

3. Write program which have complexity $n(\log n)$, n^3, $\log(\log n)$

$$T(n) = \underline{O(n \log n)}$$

```
void quicksort (int arr[], int low, int high)
{
    if (low < high)
    {
        int pi = partition (on, low, high)
        quicksort (arr, low, pi-1);
        quicksort (arr, pi+1, high);
    }

}
int partition (int arr[], int low, int high)
{
        int pivot = arr [high];
        int i = (low-1);
        for ( int J = low ; J <= high-1 ; J++)
        {
            if (arr [i] <pivot)
            {   i++;
                swap (& arr [i], & arr [J]);
        }
        swap (& arr [i+1], & arr [high])
            return (i+1);

}
```

$T(n)=n^3$

multi. of square matrix

```
for (j=0; j<d; j++)
{   for (k=0; k<d; k++)
    {   res [i][j]+= a[i][k]+b[k][j];
    }
}
```

$T(n)=O(\log(\log n))$

```
for (i=2; i<n; i=i*i)
{
        count ++;
}
```

6. Arrange the following in increasing order of rate of growth:

a) n, $n!$, $\log n$, $\log\log n$, root(n), $\log(n))$, $n\log n$, $\log^2 n$, 2^n, 2^{2^n}, 4^n, n^2, 100

$100 < \log\log n < \log n < (\log n)^2 < \sqrt{n} <$

$n\log n < \log n! < n^2 < 2^n < 4^n < 2^2$

b) $2(2^n)$, $4n$, 2^n, 1, $\log(n)$, $\log(\log(n!))$,
$\sqrt{\log n}$, $\log 2n$, $2\log(n)$, $n\log(n!)$, $n!$,
$n2$, $n\log(n)$

$1 < \log\log n < \sqrt{\log n} < \log n < \log 2^n < 2\log n <$
$n < n\log n < 2n < 4m < \log(n!) < n2 < n! < 2^{2^n}$

c) 8^{2n}, $\log_2 n$ ~~g~~ $n\log_6 n$, $n\log_2(n)$, $\log(n!)_3$
$\log_8(n)$, 96, $8nx$, $7n3$, $5n$,

$96 < \log_8 n < \log 2n < 5n < n\log_6 n < n\log_2 n <$
$\log(n!) < 8n2 < 7n3 < n! < 8^{2n}$