Tutorial - 2

Name - Divyanshu Kaira
Section - F
Roll No - 52
Univ Roll No - 2016740

Q1 What is the time complexity of below code and how?

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

↳ $j=1$ $i=$
$=2$ $i=1+2$ m-level
$j=3$ $i=1+2+3$

for(i)

$\because 1+2+3+\ldots+<n$

$\because 1+2+3+m<n$

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

$$m=\sqrt{n}$$

By summation method

$\sum_{i=1} 1 \Rightarrow 1+1+\ldots+\sqrt{n}$ times

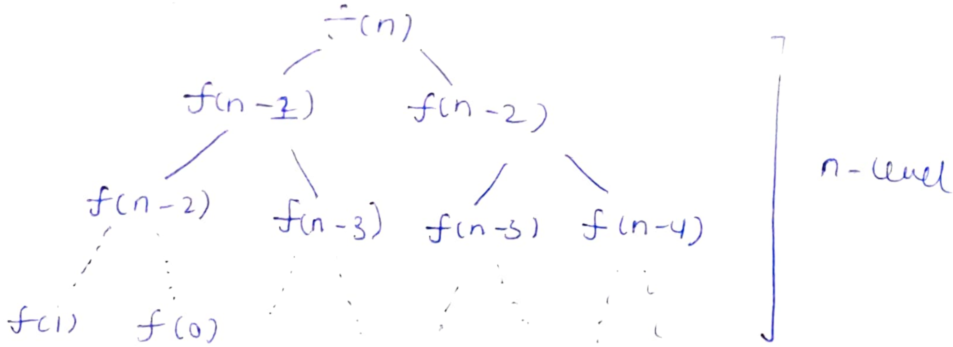
$\boxed{T(n)=\sqrt{n}}$ — Ans

Q2 Write recurrance relation for function that prints Fibonacci series. Solve it to get the time complexity. What will be the space complexity and why?

→ For Fibonacci series

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

By forming a tree $f(0)=0$
$f(1)=1$



∵ At every function call we get = Function calls
∴ for n levels
we have = 2×2 ...... n times

$\boxed{\therefore T(n) = 2^n}$

MAXIMUM SPACE

Considering Recursive

Stack:
no. of calls maximum = n

For each call we have space complexity O(1)

$\boxed{\therefore T(n) = O(n)}$

without considering Recursive Stack:
each call we have time complexity O(1)

$\boxed{\therefore T(n) = O(1)}$

Q3 Write program which have complexity:
n(log n), n^3, log(log n)

1) nlogn → Quick Sort

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

} }
```

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

2 $n^3 \rightarrow$ Multiplication of 2 square number matrix

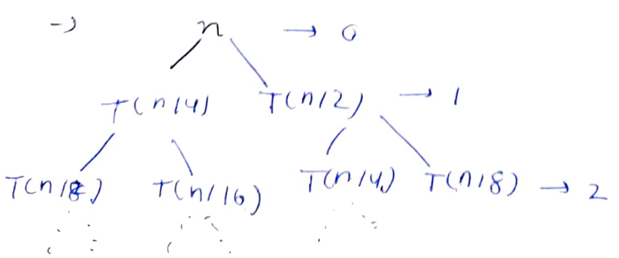
```
for (int i=0; i<n; i++)
{
  for (j=0; j<c2; j++)
  { for (k=0; k<c1; k++)
    { res[i][j] += a[i][k] * b[k][j];
    }
} }
```

3) log (log n)

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

Q4 Solve the following recurrence relation

$T(n) = T(n/4) + T(n/2) + cn^2$



At level

$0 \rightarrow Cn^2$

$1 \rightarrow \frac{n^2}{4^2} + \frac{n^2}{2^2} = \frac{C5n^2}{16}$

$2 \rightarrow \frac{n^2}{8^2} + \frac{n^2}{16^2} + \frac{n^2}{4^2} + \frac{n^2}{8^2} = \left(\frac{5}{16}\right)^2 n^2 C$

max level $= \frac{n}{2^K} = 1$

$= K = \log_2 n$

$T(n) = C(n^2 + 5/16)n^2 + (5/16)^2 n^2 + \dots + (5/16)^{\log n} n^2)$

$T(n) = Cn^2 [1 + (\frac{5}{16}) + (\frac{5}{16})^2 + \dots + (\frac{5}{16})^{\log n}]$

$T(n) = Cn^2 \times 1 \times \left(\frac{1 - (5/16)^{\log n}}{1 - (5/16)}\right)$

$T(n) = Cn^2 \times \frac{11}{5} \times (1 - (\frac{5}{16})^{\log n})$

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

$O(n^2)$ Ans

Q.5> What is the time complexity of fallowing fun()?

```
int fun(int n)
{
  for(int i=1; i<=n; i++)
  {
    for(int j=1; j<n; j+=i)
    {  Some task O(1)
  } }
}
```

c) 8^{2n}, $\log(n)$, $n\log_6(n)$, $n\log_2(n)$, $\log(n!)$, $n!$, $\log_8(n)$,

96, $8n^2$, $7n^3$, $5n$

→ $96 < \log_8 n < \log_2 n < 5n < n\log_6(n) < n\log_2 n <$

$\log(n!) < 8n^2 < 7n^3 < n! < 8^{2n}$

---

d)