Tutorial - 2

Name - Amandeep Singh
Section - G
Roll No. - 53
Uni Roll No. - 2016623

Course - B. Tech CSE
Semester - 4

Q1

Ans 1

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

$\Big]$ m - level

for (i)

$\because \; 1+2+3+4 \ldots + < n$

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

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

$m \approx \sqrt{n}$

By Summation method :-

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

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

Q2

Ans 2 for fibonacci Series :

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

$\because$ At every function call, we get 2 other function call.

∴ for n levels

we have = 2×2 . . . n times

$T(n) = 2^n$

* <u>Maximum Space</u>
↳ Considering recursion.

Stack: no. of calls max = n
for each call we have space complexity O(1)
∴ $T(n) = O(n)$

→ without considering recursion stack:
each call have time complexity (O(1))
∴ $T(n) = O(1)$

Q3

Ans 3 (i) n logn → 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 pivot = arr [high];
    int i = (low-1);

  for (int j=low ; j <= high ; j++)
   {
     if (arr [j] < pivot)
      {
```

```
        i++;
        swap (&arr[i], &arr[j]);
      }
   }
   swap (&arr[i+1], &arr[high]);
   return (i+1);
}
```

(2) n^3 → Multiplication of 2 Square Matrix

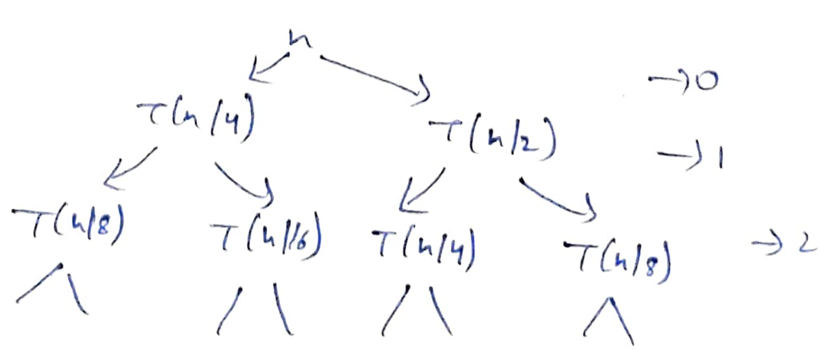
```
for (i=0; i<r1; i++)
{
   for (j=0; j<c2; j++)
      for (k=0; k<c1; k++)
      {
         res[i][j] += a[i][k] * b[k][j];
      }
}
```

(3) log (log n)

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

Q4

Ans 4



At level

$0 \rightarrow cn^2$

$1 \rightarrow \frac{n^2}{4^2} + \frac{n^2}{2^2} = \frac{5n^2}{16} c$

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

$\vdots$

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

$\Rightarrow k = \log_2 n$

$T(n) = c\left(n^2 + (5/16) n^2 + (5/16)^2 n^2 + \dots + (5/16)^{\log n} \cdot n^2\right)$

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

$T(n) = cn^2 * 1 * \left[\frac{1 - (5/16)^{\log n}}{1 - (5/16)}\right]$

$T(n) = cn^2 * \frac{11}{15} * \left[1 - \left(\frac{5}{16}\right)^{\log n}\right]$

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

$\underline{O(cn^2)}$ Ans

Q5

Ans 5: for

i	j
1	1
2	1 + 3 + 5
3	1 + 4 + 7
$\vdots$	$\vdots$
n	$\vdots$

$j = (n-i)/i$ times

$$\sum_{i=1}^{n} \left(\frac{n-1}{i}\right)$$

$$T(n) = \left(\frac{n-1}{1}\right) + \left(\frac{n-1}{2}\right) + \left(\frac{n-1}{3}\right) \cdots + \frac{(n-1)}{n}$$

$$T(n) = n\left\{1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}\right] - 1 \times \left[1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}\right]$$

$$= n \log n - \log n$$

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

Q 6

Ans 7 for

i

2^1

2^k

2^{k^2}

2^{k^3}

$\vdots$

2^{k^m}

where

$2^{k^m} <= n$

$k^m = \log_2 n$

$m = \log k \log_2 n$

$$\therefore \sum_{i=1}^{m} 1$$

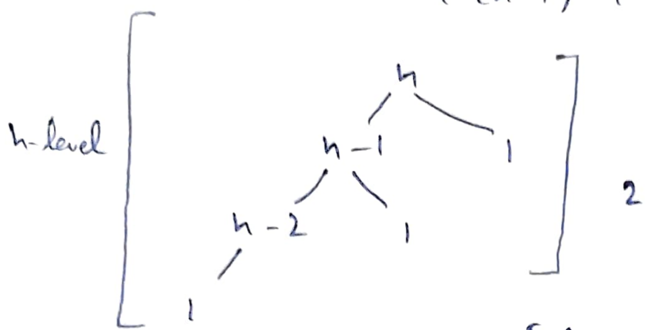
$1 + 1 + 1 + \ldots$ m times

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

Q7

Ans 7 Given, Algo. divides array in 99% & 1% part.

$\because T(n) = T(n-1) + O(1)$



'n' work is done at each level

$T(n) = (T(n-1) + T(n-2) + \ldots + T(1) + O(1)) * n$

$= n * n$

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

lowest height $= 2$

highest height $= n$

$[\because$ difference $= n-2]$ $n > 1$

$\Rightarrow$ The given Algo produces linear result

Q8

Ans 8 (a) $100 < \log\log n < \log n < (\log n)^2 < \sqrt{n} < n < n\log n < \log(n!) < n^2 < 2^n < 4^n < 2^{2^n}$.

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

(c) $96 < \log_8 n < \log 2n < 5n < n\log(n) < n\log n < \log(n!) < 8n^2 < 7n^3 < n! < 8^{2^n}$.