CST
Roll no → 29
Tushar Sharma

Data / /
Page No.

DAA- Assignment

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

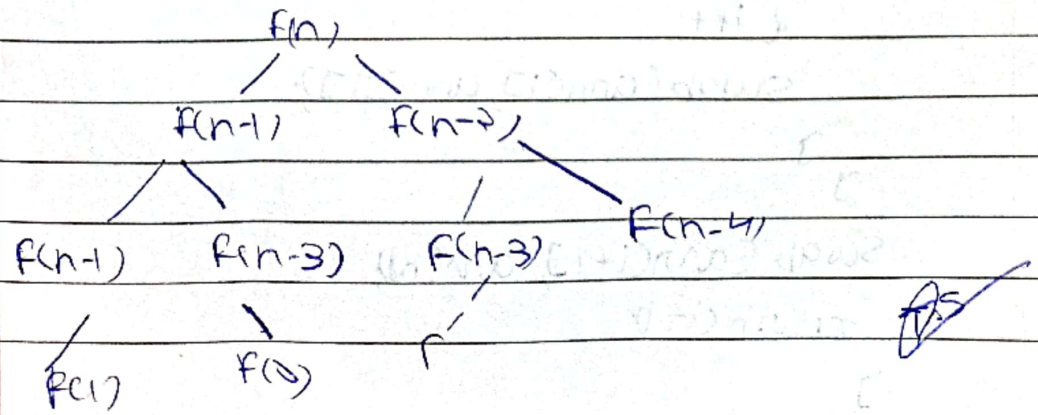
for J=1 i=1
J=2 i=1+2
J=3 i=1+2+3
} n levels

∵ 1+2+3+ - - - +<n
1+2+3+...n <n
$\frac{n(n+1)}{2} < n$

$n \approx \sqrt{n}$

by summation method
1 = 1+1+ - - - $\sqrt{n}$ times
∴ $T(n) = \sqrt{n}$

② For fibbonacci series
$f(n) = f(n-1) + f(n-2)$
$f(0) = 0 \quad f(1) = 1$



Date / /
Page No.

At every function call we get two function calls
for n levels
we have = 2x2 ---n times
$T(n) = 2^n$

minimum space:- considering recursive
Stack:-
no of calls max = n
for each call we have space complexity O(1)
$\therefore T(n) = O(n)$

(3) a) n log n

```
quick sort (int arr[], int l, int h)
{ if (l<h)
  { int pi = portion (arr, l, h);
  func(arr, l, pi-1)
  func(arr, pi+1, h);
  }
}

int portion (int arr[], int l, int h)
{ int pi = arr[h];
    int i = (l-1);

for (int j=l; j<=h; j++)
  { if (arr[j] < pi]
      { i++;
      swap[arr[i], arr[j]];
    }
  }
  Swap [arr(i+1], arr(h]);
  return (i+1);
}
```

(b) n^3

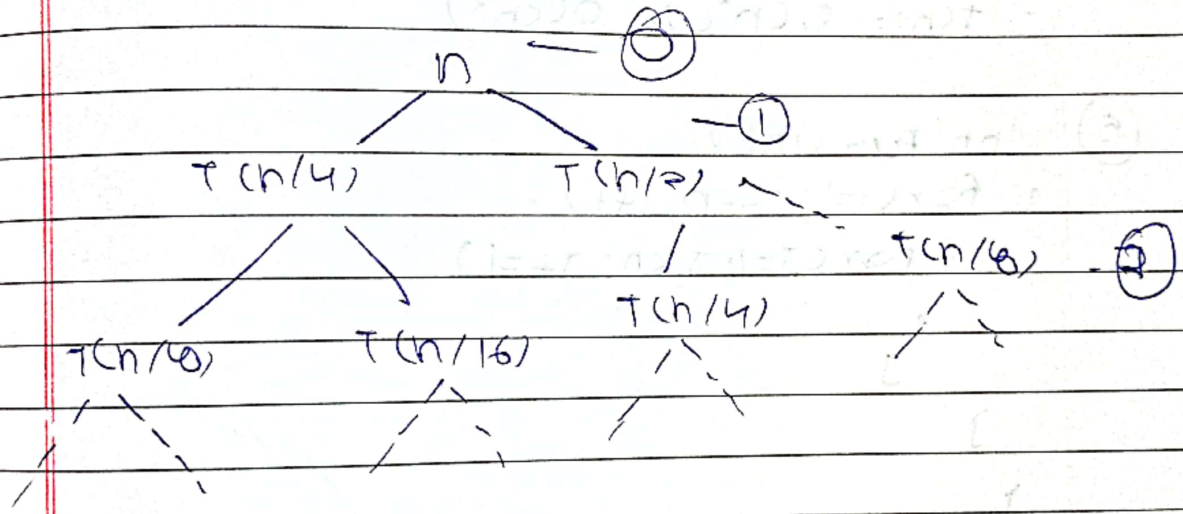
multiplication of two square matrix

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

(c) log(logn)

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

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



At level $\rightarrow O(n^2)$

$1 \rightarrow \frac{n^2}{4} + \frac{n^2}{2^2} = \frac{c5n^2}{16}$

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

T.S

max level $= \frac{n}{2^k} = 1 \Rightarrow k = \log_2 n$

$\therefore T(n) = \left[n^2 + \left(\frac{5}{16}\right) n^2 + \left(\frac{5}{16}\right)^2 + \cdots \left(\frac{5}{N}\right) \log^n n^2 \right]$

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

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

$= cn^2 \times \frac{11}{5} \left(1 - \left(\frac{5}{16}\right) \log n\right)$

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

(5)

```
int fun(int)
{ for(i=1; i<=n; i++)
    { for(J=1; J<n; J+=i)
        {
          J
        }
    J
}
```

for i	J
1	1
2	1+3+5
3	1+4+7
4	1+5+9
⋮	⋮
n	log n

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

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

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

$= n \log n - \log n$

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

(6) for (i=2 ; i<=n ; i = pow(i,k))
{ O(1)
}

for $\Rightarrow$ i
2^1
2^k
2^{k^2}
2^{k^3}
1
2^{k^m}

where, $2^{k^n} <= n$

$k^m = \log_2 n$

$m = \log_k \log_2 n$

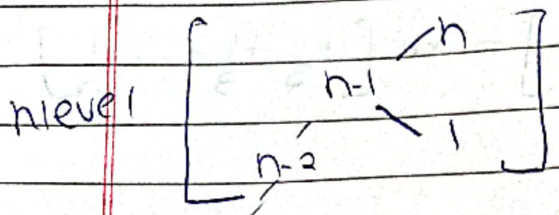
$\sum_{i=1}^{m} = 1$

$\Rightarrow 1 + + \text{ n times} = T(n) = O(\log_k \log_2 n)$

Date / /
Page No.

(7) given algo divides array in 99% & 1% part

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



'n' work is done at each level for merging

$T(n) = [T(n-1) + T(n-2) + \cdots T(1) + O(1) \times n]$

$\Rightarrow n \times n$

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

lowest height $= 2$

highest height $= n$

$\therefore$ diff $= n-2 \quad \because (n > 1)$

(8) considering for large value of 'n'

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

(b) $1 < \log\log n < \sqrt{\log n} < \log n < \log_2 n < 2\log n < n < n\log n < 2n < 4n < \log(n!) < n^2 < n < 2^{\log_2 n} < 5n$

(c) $96 < \log_6 n < \log 2n < 5n < n\log_6 n < n\log_2 n < \log(n!) < 6n^2 < 7n^3 < n! < 6^2 n$