Design and Analysis Of Algorithm

Tutorial 2

Name – Shikha , Section – C

Roll No – 2014853

Q.1

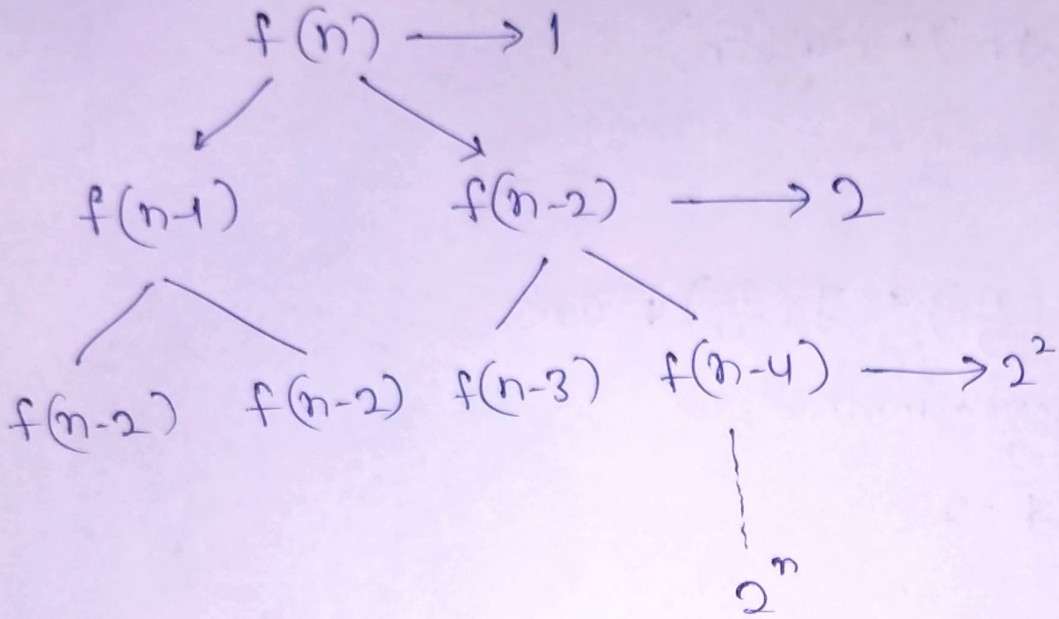
```
void fun(int n)
{
  int j=1, i=0;
  while (i<n) {
    i = i+j;
    j++;
  }
}
```

j=1	i=0
1	1
2	3
3	6
4	10

Series of i nearly dependent on j as 2^j

Time Complexity = $O(2^n)$

Q.2 Space Complexity = $O(n)$ as clear call of $f(n-1)$



Time Complexity = $O(2^n)$

Q.3 i) $n(\log n)$

```
for(i=0; i<n; i++)
for(j=0; j<n; j=j*2)
  c++;
```

ii) n^3

```
for(i=0; i<n; i++)
for (j=0; j<n; j++)
for (k=0; k<n; k++)
  c++;
```

iii) $\log(\log n)$

```
int fun(int n)
{ if(n==1)
  return n;
```

```
else
return fun(√n) + fun(√n)
}
```

Q4. $T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + cn^2$

using master's method:

$a = 2$, $b = 2$

$c = 1$

$f(n) > n^2$ $\quad$ $f(n^2) > 1$

Time Complexity = $O(n^2)$

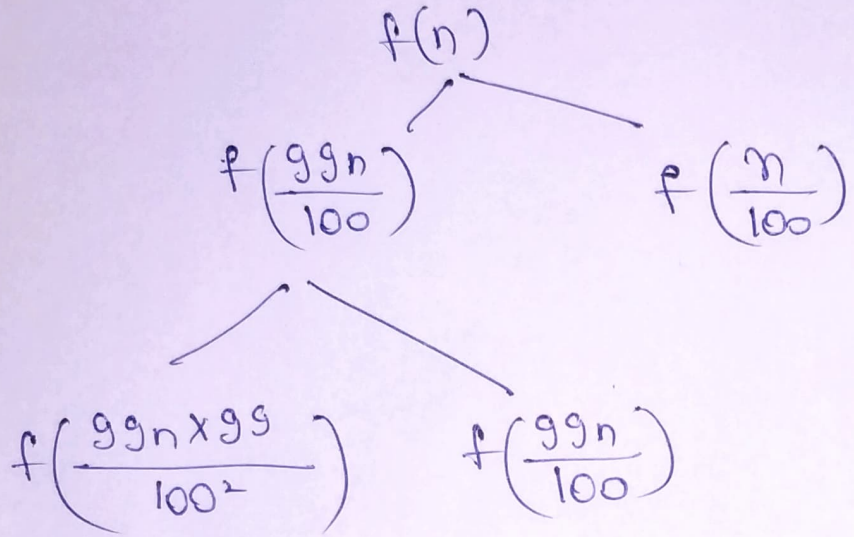
Q5.
```
int fun(int n)
for(int i=1; i<=n; i++) {
for(int j=1; j<n; j=j+i)
{
// some O(1) task
}
}}}
```

Time Complexity : $O(n\sqrt{n})$

Q6.
```
for(int i=2; i<=n; i=pow(i,k))
{
    // some O(1) statements or expressions
}
```

where k is a constant (Time Complexity - $O(\log(\log(n)))$

Q.7. $T(n) = T\left(\frac{99n}{100}\right) + T\left(\frac{n}{100}\right)$



Time Complexity - $O(\log n)$

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

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

c) $96 < \log_2 n < \log_3 n < \log 5n < \log n! < n\log_6(n) < n\log_2(n) < 8n^2 < 7n^3 < 3^{2n} < n!$

Shikha