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

i	j
0	1
1	2
3	3
6	4
1+2+3+...+k	k

$1+2+3+4+5+\ldots+k = n$ → it is an AP.

$$\frac{k*(k-1)}{2} = n$$

$$k^2 = n$$

$k^2 < n$ → termination condition

$k = \sqrt{n}$

2. int fib (int n) → T(n)
{
if (n<=1) → O(1)
return n;
else
return fib(n-1) + fib(n-2); → T(n-1) + T(n-2)
}

$T(n) = T(n-1) + T(n-2) + 1$ (we will a assume $T(n-1) = T(n-2)$)

$T(n) = 2T(n-1) + 1$ — ①

$T(n-1) = 2T(n-2) + 1$ — ②

Putting ② in ①

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

$= 4T(n-2) + 2 + 1$ — ③

now $n = n-2$ in ①

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

$= 2T(n-3) + 1$ — ④

④ in ③

$T(n) = 4(2T(n-3)+1) + 2 + 1$

$= 2^3 T(n-3) + 4 + 2 + 1$

$T(n) = 2^k (T(n-k) + 1 + 2 + 4 + \ldots 2^{k-1}$

$n-k = 0$

$n = k$

$= 2^n (T(n-n)) + 2^n - 1$

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

Space complexity → O(n)

Ques - 3

(i) $n(\log n)$

```
int main()
{ int count=0; int n ; cin>>n;
for (int i=0 ; i<n ; i++)          ——— n
    {
    for (int j= 0; j <n ; j=j*2 )  ——— (n-1) * log n
      {
        count ++;
      }
    }
    cout << "count" << count ;
}
```

(ii) n^3

```
int main()
{ int count=0 ; int n ;
  cin >>n ;
  for (int i=0 ; i<n ; i++)          — n
    {for(int j=0 ; j<n ; i++)        — n * (n-1)
      { for(int k=0 ; k<n ; i++)     — n * (n-1) * (n-1)
        {
          cout<< "*";
        }
      }
  }
```

(iv) $\log(\log n)$

```
int main()
{ int count=0; int n;
  cin>>n;
  for (int i=0; i<n; i=i*2)          — log2 n
    {for(int j=0; j<n; j=j*2)        ——— log (log2 n)
      {
        count ++;
      }
    }
}
```

Ques4 → $T(n) = T(n/2) + Cn^2$ — ① replacing $T(n/4)$

$T(n/2) = T(n/4) + C\frac{n^2}{2^2}$ — ② ② in ①

$T(n) = T(\frac{n}{4}) + C(\frac{n}{2})^2 + Cn^2$ — ③

$T(n/4) = T(\frac{n}{8}) + C(\frac{n}{4})^2$ — ④

④ in ③

$T(n) = T(\frac{n}{8}) + C(\frac{n}{4})^2 + C(\frac{n}{2})^2 + C(n)^2$

$T(n) = T(\frac{n}{2^k}) + Cn^2(1 + \frac{1}{4} + \frac{1}{16} + \cdots + \frac{1}{2^{2k}})$

$n = 2^k$

$T(1) + Cn^2\left[\frac{1 \times (1-(\frac{1}{4})^k}{1-\frac{1}{4}}\right]$

$1 + Cn^2\left[\frac{4^k - 1}{\frac{4^k}{3}}\right]$

$\Rightarrow O(n^2)$

Ques5 -
```
int fun (int n) {
  for (int i=1; i<=n; i++) ——— n
    { for(int j=1; j<n; j+=i) ———
      //some O(1)
    }
}
```

i	j
1	1, 2, 3, 4 . . . n
2	1, 3, 5 . . . n/2
3	1, 4, 7 . . . n/3

Total complexity $\Rightarrow n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \cdots + \frac{n}{k}$

$n(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{k})$

$O(n[\log n])$

Ques6 →

$2, 2^k, 2^{2k}, \ldots, 2^{ki}$

$2^{ki} = n$ —— taking log both side

$\log 2^{ki} = \log_2 n$

$ki \log_2 2 = \log_2 n$

$ki = \log n$ —— taking log both side

$i \log k = \log(\log n)$

$i = \log_k \log n$ — Time complexity

Q3→

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

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

(c) $96 < \log_3 n < \log_2 n < \log_{10} n < 5n < \log n! < n\log_6 n < n\log_2 n < 8n^2$
$7n^3 < n! < 8^{2n}$