Tutorial-2

① What is the time complexity of below Code and how?

```
Void fun(int n)
{ int j= 1, i= 0;
    while (i<n)        →   0 to n times
    { i= i+j;          →   n times
      j++;             →   n times
    }
}
```

Ans O(n)

② Write Program which have Complexity —
n (logn), n³, log (logn).

Ans n (logn) → Quick sort

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

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);
 } }
```

n^3 → ~~for~~ 3 variable eqn

```
function find xyz(n)
   { const solution = [];
  for(~~lett~~ n = 0; n<n; x++)
    { for (int y=0; y<n; y++)
      { for (int z=0; z<n; z++)
        { if (3*n + 9*y + 8*z = 79).
          { Solution. Push ({n, y, z});
       }
    }
         return solutions;
  }
```

log(n)

```
# include <iostream>
   using namespace std;
  unsigned int log2n(unsigned int n).
     { return (n>1) ? 1 + log2n(n/2) : 0;
     }

     int main()
       { unsigned int n=32;
            cout<< log2n(n);
            getchar();
              return 0;
          }
```