

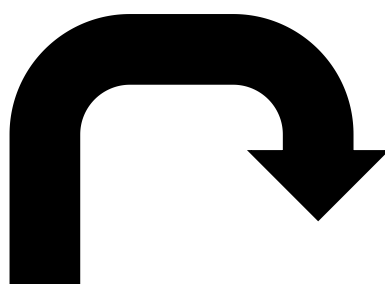
# Assignment DAA

Ananya(590013832)

- Binary Search Algorithm:  
Function for Binary Search can be :

```
int binarysearch(int arr[ ],int n,int key) {  
    int low=0;  
    int high=n-1;  
    int mid;  
    While(low<=high){  
        Mid=(low+high)/2;  
        If(arr[mid]==key)  
            return mid;  
        else if(arr[mid]>key)  
            Low=mid+1;  
        else  
            high=mid-1;  
    }  
}
```

- CODE PROOF:



Code proof :-

$$T(n) = k + T(n/2)$$

$$T(n) = T(n/2) + k \quad \text{--- (1)}$$

$$T(n/2) = T(n/4) + k \quad \text{--- (2)}$$

$$T(n/4) = T(n/8) + k \quad \text{--- (3)}$$

⋮

$$T(n/2^{k-1}) = T(n/2^k) + k$$

add (1) (2) (3) ...

$$T(n) = T\left(\frac{n}{2^k}\right) + ck \quad \text{--- (A)}$$

here  $\frac{n}{2^k} = 1 \quad \therefore$  at last only one element remains.

So,

$$n = 2^k$$

$$\boxed{k = \log_2 n} \quad \text{hence proved}$$

- Prove that value of  $k$  denotes time taken by binary search algo.

MAY MONDAY 08  
\* Prove that value of  $k$  denotes time taken by binary search algo.

⇒

last page

put  $k = \log_2 n$  in (A)

$$T(n) = T\left(\frac{n}{2^{\log_2 n}}\right) + C \log_2 n$$

$$\text{here } 2^{\log_2 n} = n$$

$$T(n) = T\left(\frac{n}{n}\right) + C \log_2 n$$

$$T(n) = T(1) + C \log_2 n$$

here  $T(1)$  is almost negligible  $T(1) = 0$

$$T(n) = C \log_2 n$$

So indirectly, directly

$$\boxed{T(n) = k = \log_2 n}$$