It-081 Pankhania Aanandi R.

DAA : Binary Search

→ half-interval or logarithmic search

→ Data should be in array & sorted

Algo :-

1.) Set L to 0 and R to n-1.

[L - left most element's index
R - right most " "]

2.) If L > R, the search terminates as unsuccessful.

3.) Set m (the position of the middle element) to the floor (the largest previous integer) of (L+R)/2.

4.) if T = Am the search is done.
→ Return m.

5.) if T > Am set L to m+1 & go to step 2.

6.) if T < Am set R to m-1 & go to step 2.

```
int binarySearch (int arr[],
                  int l, int r,
                  int x)
{
    if (r >= 1)
    {
        int mid = l + (r-1) / 2 ;

        if (arr[mid] == x)
            return mid;

        if (arr[mid] > x)
            return binarySearch (arr, l,
                                 mid-1, x);

        return binarySearch (arr, mid+1, r,
                             x);
    }
    return -1;
}
```

SUNDAY 31

WK 13 • 090 275

Analysis:-

Array: { 1, 2, 3, 50, 60, 81, 4, 123, 51, 40, 86 }
0 10

enter element : 2 (=x)

1. ⟹ l = 0 , r = 10

mid = 0 + (10-1)/2 = 4

arr[mid] = 60

⟹ 2 < 60

2. ⟹ l = 0 r = 3

mid = 0 + (3-1)/2 = 1

arr[mid] = 2

⟹ 2 = 2

return 2.