

0	1	2	3	4	5	6	7	8	9
5	15	20	25	30	35	40	45	50	55

```

int linearSearch (int[] arr, int t) {
    for (i=0 ; i < arr.length; i++) {
        if (arr[i] == t) {
            return i;
        }
    }
    return -1;
}

```

if  $t = 5$   
 $O(1)$

if  $t = 55$   
 $O(10) = O(n)$

if  $t = 49$   
 $O(10) = O(n)$

arr	0	1	2	3	4	5	6	7	8	9
	5	15	20	25	30	35	40	45	50	55

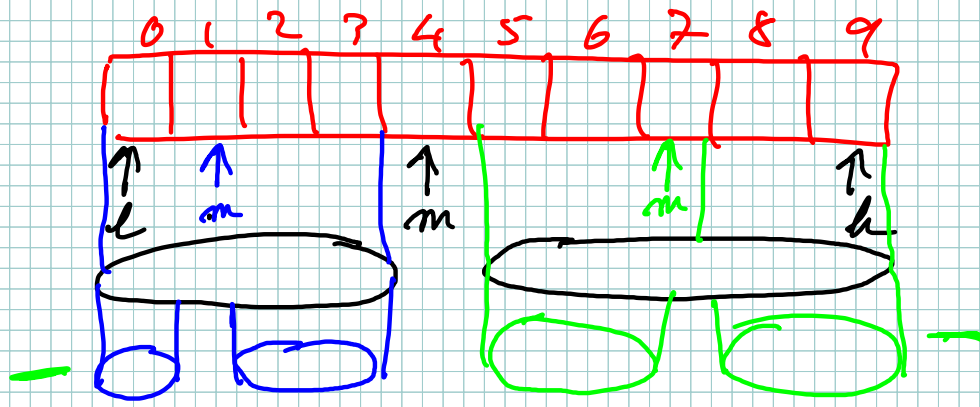
int binarySearch (int[] arr, int t) {

int l = 0;  
int h = arr.length - 1;  
int mid;

l	h	mid	t
0	9	4	40
5	7	7	
8	8	8	

- while (h >= l) {  
- mid = (l + h) / 2;  
- if (arr[mid] == t) {  
return mid;  
} else if (arr[mid] > t) {  
h = mid - 1;  
} else {  
l = mid + 1;  
}  
}  
return -1;

}



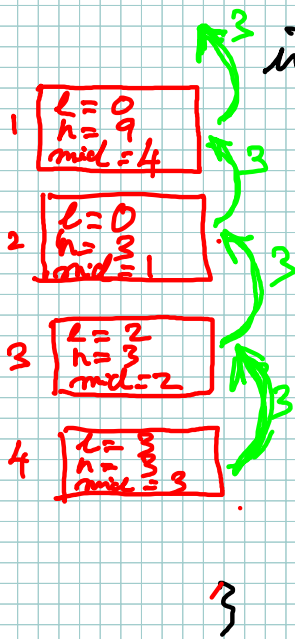
$$\log n < n$$

$$O(\log_2 n)$$

$$O(\log n)$$

0	1	2	3	4	5	6	7	8	9
5	10	15	20	25	30	35	40	45	50

t = 20



```

int binarySearch (int arr, int t,
                 int l, int h) {
    if (h < l) {
        return -1;
    }
    int mid = (l+h)/2;
    - if (arr[mid] == t) {
        return mid;
    }
    - else if (arr[mid] > t) {
        return binarySearch (arr, t, l, mid-1);
    }
    - else {
        return binarySearch (arr, t, mid+1, h);
    }
}

```

$O(\log n)$