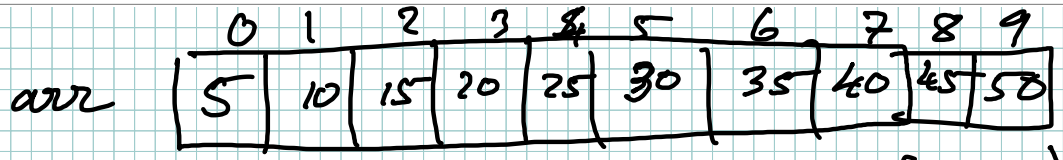
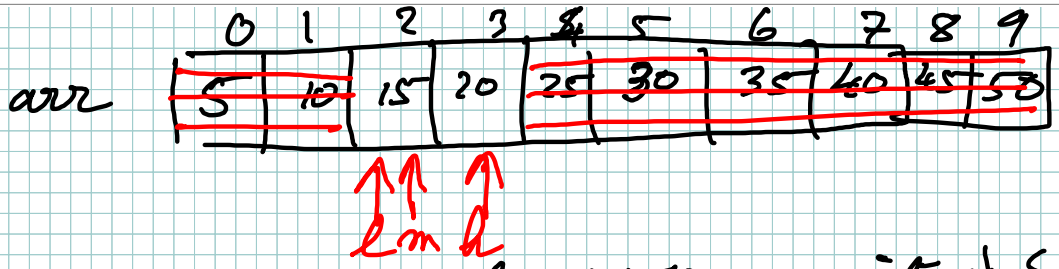



```
int linearSearch(int[] arr, int t) {
    for (i=0; i<arr.length; i++) {
        if (arr[i] == t) {
            return i;
        }
    }
    return -1;
}
```

not found O(n)

t = 5 O(1)



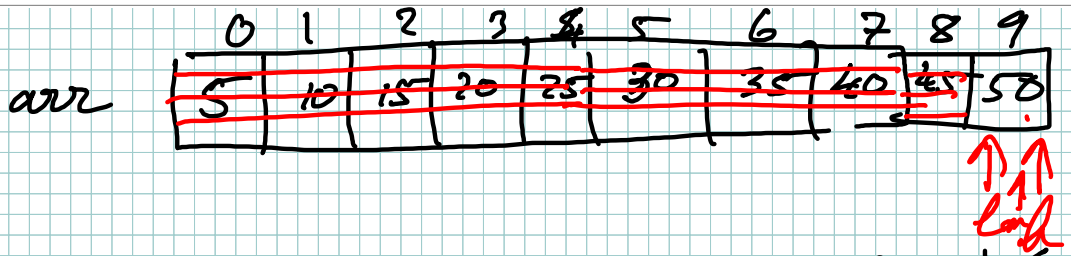
t = 15

l	h	m
0	9	4
2	3	1
		2

```
int binarySearch(int[] arr, int t) {
    int l = 0;
    int h = arr.length - 1;
    int m;
    while (h >= l) {
        m = (l + h) / 2;
        if (arr[m] == t) {
            return m;
        } else if (arr[m] > t) {
            h = m - 1;
        } else {
            l = m + 1;
        }
    }
    return -1;
}
```

$O(\log_2 n)$

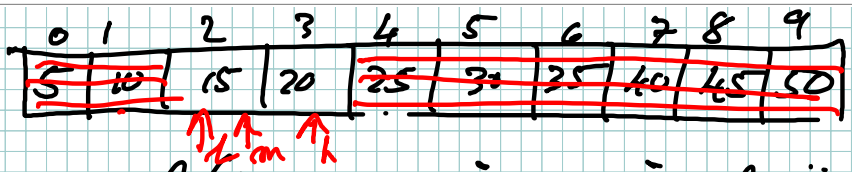
$O(\log n)$



t=49

l	h	m
0	9	4
5	8	7
8		8
9		9

```
int binarySearch(int[] arr, int t){
    int l = 0;
    int h = arr.length - 1;
    int m;
    while(h >= l){
        m = (l + h) / 2;
        if(arr[m] == t){
            return m;
        }else if(arr[m] > t){
            h = m - 1;
        }else{
            l = m + 1;
        }
    }
    return -1;
}
```




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