

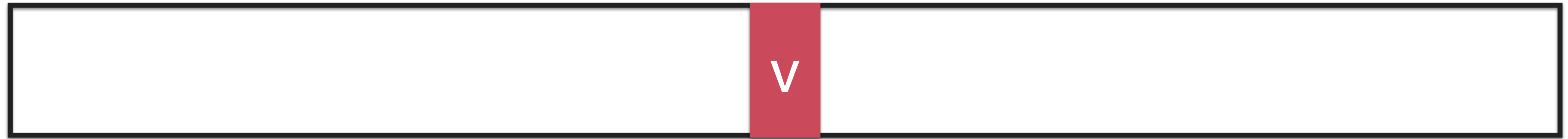
算法与数据结构体系课程

liuyubobobo

更多关于二分查找法

二分查找法的变种：upper

二分查找法



查找存在或者不存在

查找大于 target 的最小值

二分查找法

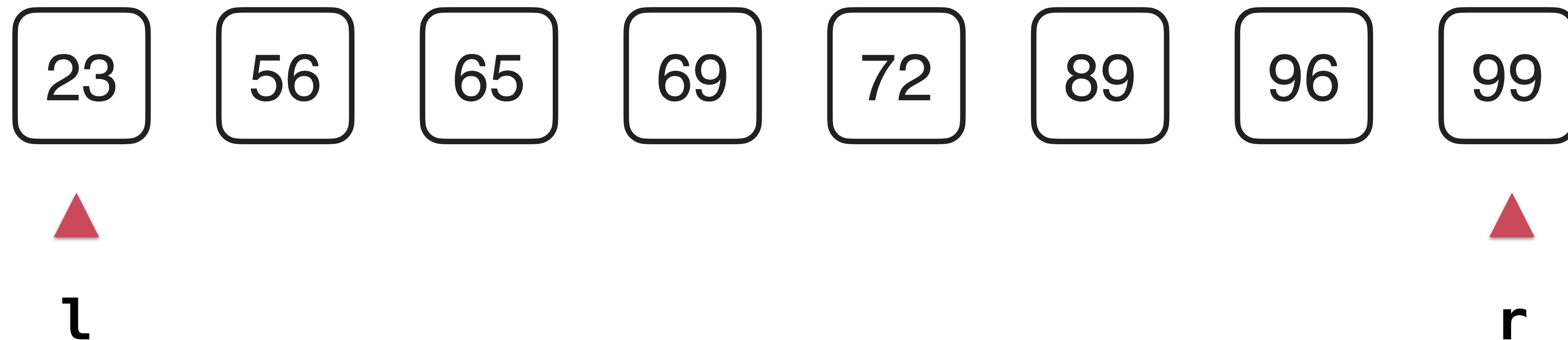
查找大于 target 的最小值

查找大于 60 的最小值

23	56	65	69	72	89	96	99
----	----	----	----	----	----	----	----

二分查找法

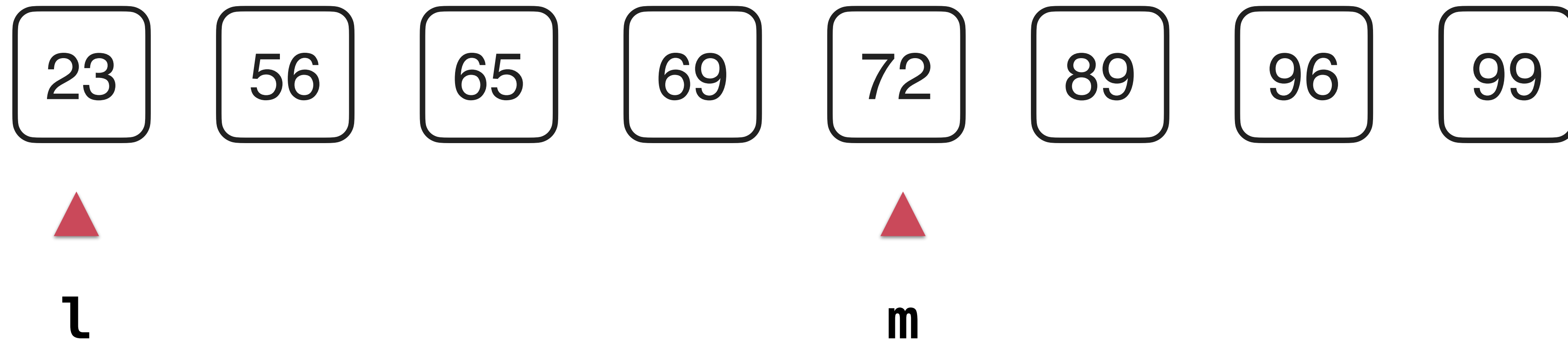
查找大于 target 的最小值 搜索范围 arr[l, r]



二分查找法

查找大于 target 的最小值 搜索范围 $\text{arr}[l, r]$ $r = \text{arr.length}$

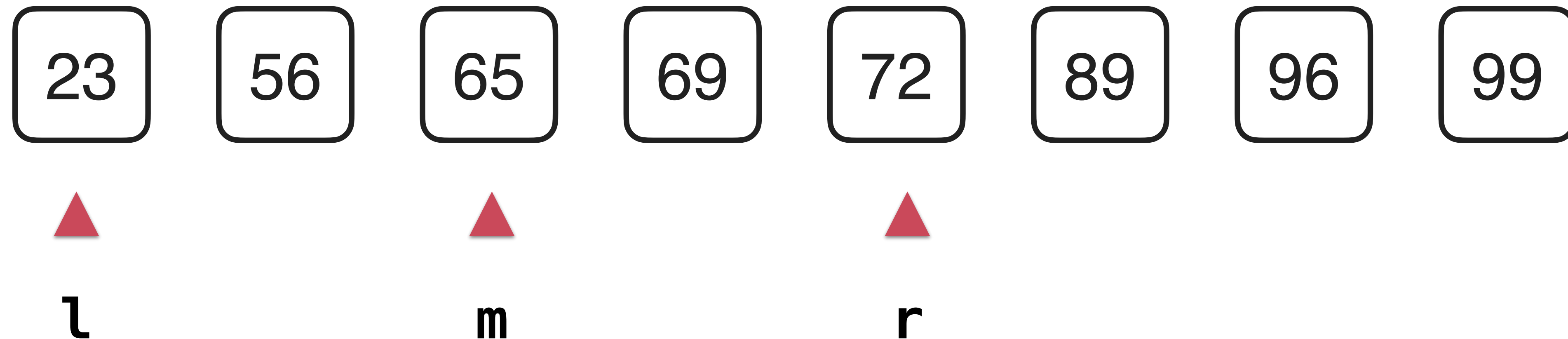
$\text{if}(\text{arr}[\text{mid}] > \text{target}) \text{ r} = \text{mid}$



二分查找法

查找大于 target 的最小值 搜索范围 $\text{arr}[l, r]$ $r = \text{arr.length}$

$\text{if}(\text{arr}[\text{mid}] > \text{target}) \text{ r} = \text{mid}$

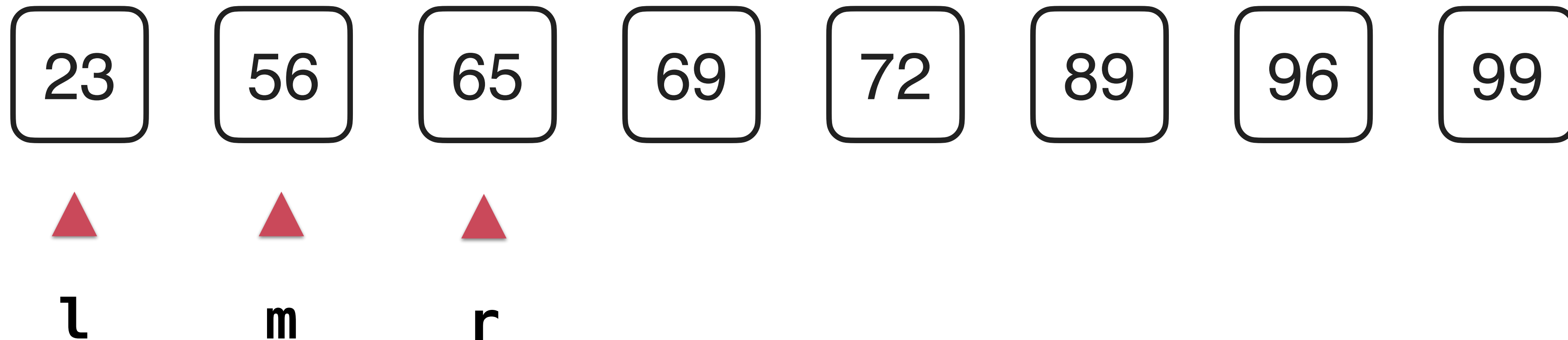


二分查找法

查找大于 target 的最小值 搜索范围 $\text{arr}[l, r]$ $r = \text{arr.length}$

$\text{if}(\text{arr}[\text{mid}] > \text{target}) \text{ r} = \text{mid}$

$\text{if}(\text{arr}[\text{mid}] \leq \text{target}) \text{ l} = \text{mid} + 1$

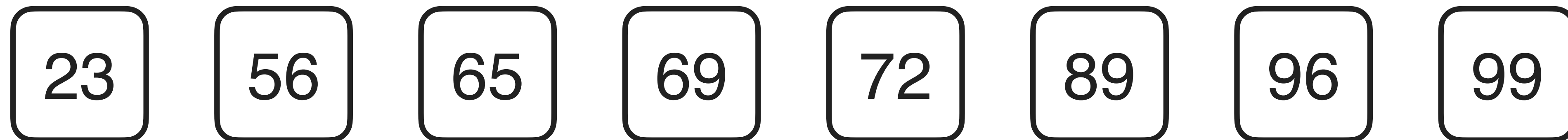


二分查找法

查找大于 target 的最小值 搜索范围 $\text{arr}[l, r]$ $r = \text{arr.length}$

$\text{if}(\text{arr}[\text{mid}] > \text{target}) \text{ r} = \text{mid}$

$\text{if}(\text{arr}[\text{mid}] \leq \text{target}) \text{ l} = \text{mid} + 1$



$\text{l} \quad \text{r}$

实现 upper

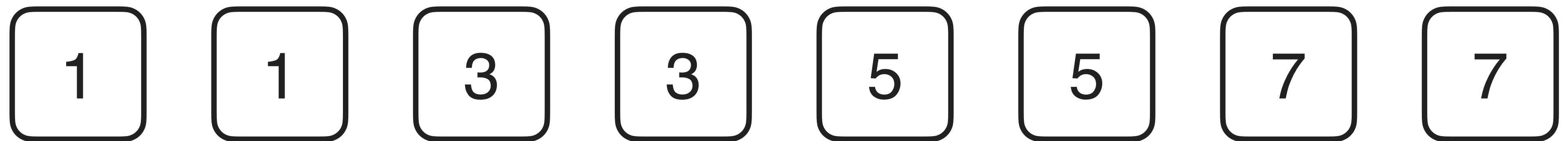
实践：实现 upper

二分查找法的变种： `ceil`

ceil

查找 5 如果数组中存在元素，返回最大索引

查找 6

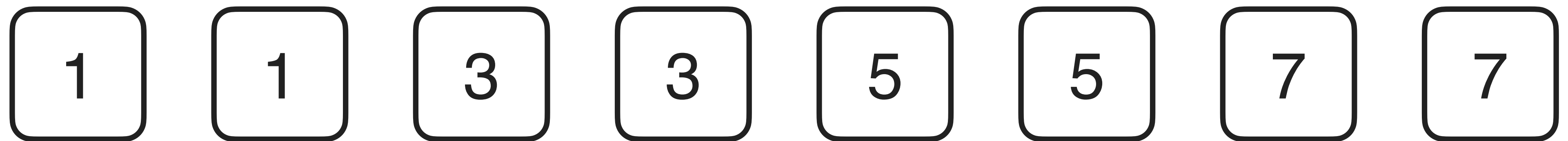


r

ceil

查找 5 如果数组中存在元素，返回最大索引

查找 6 如果数组中不存在元素，返回 upper



$\text{ceil}(5.0) = 5$ $\text{ceil}(5.5) = 6$

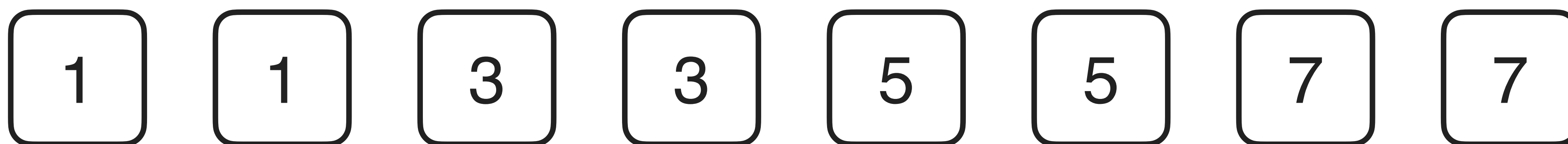
r

实践：实现 ceil

二分查找法的变种： `lower_ceil`

lower_ceil

查找 5 如果数组中存在元素，返回最大索引

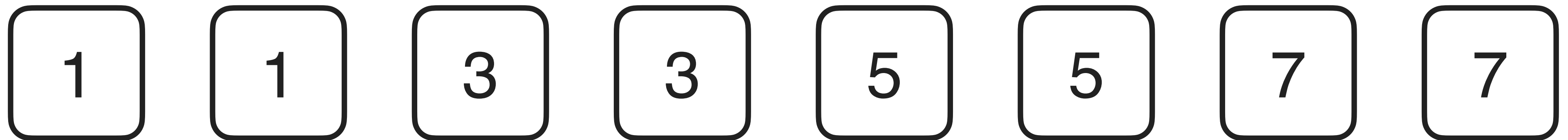


r

lower_ceil

查找 5 如果数组中存在元素，返回最小索引

查找 6 如果数组中不存在元素，返回 upper

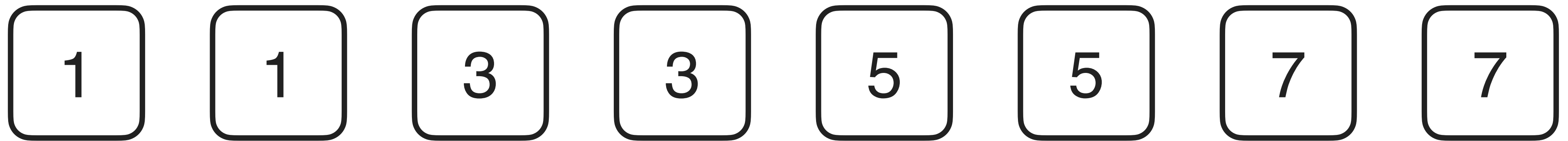


r

lower_ceil

查找 5 如果数组中存在元素，返回最小索引

查找 6 如果数组中不存在元素，返回 upper



\geq target 的最小索引



r

作业解析： 实现 lower_ceil

二分查找法的变种：lower

二分查找法

查找小于 target 的最大值

查找小于 85 的最大值

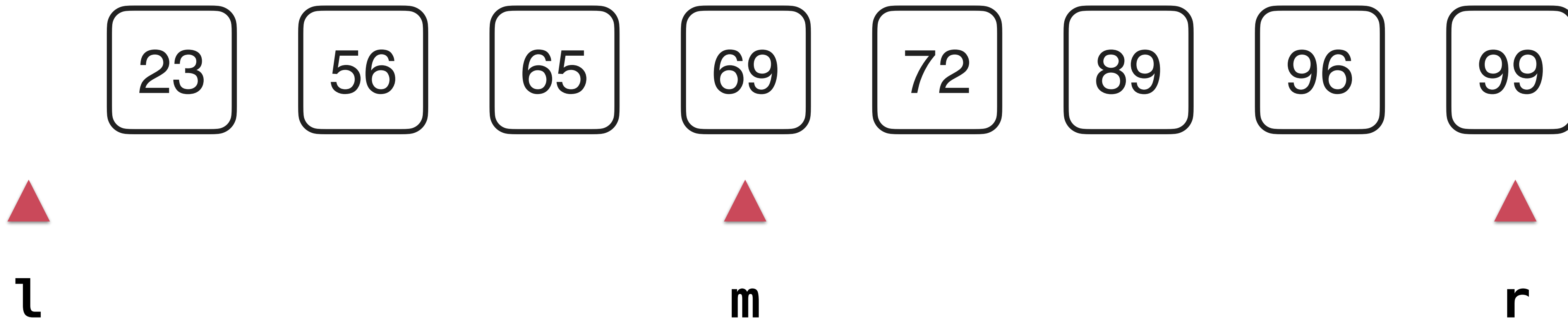
23	56	65	69	72	89	96	99
----	----	----	----	----	----	----	----

二分查找法

查找小于 target 的最大值 搜索范围 arr[l, r] l=-1

if(arr[mid] < target) l = mid

if(arr[mid] >= target) r = mid - 1



实现 lower

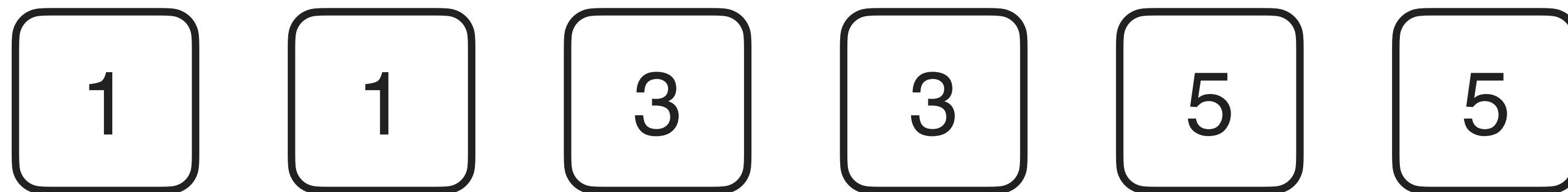
实践：实现 lower

lower 的问题

查找小于 target 的最大值 搜索范围 arr[l, r] l=-1

if(arr[mid] < target) l = mid

if(arr[mid] >= target) r = mid - 1



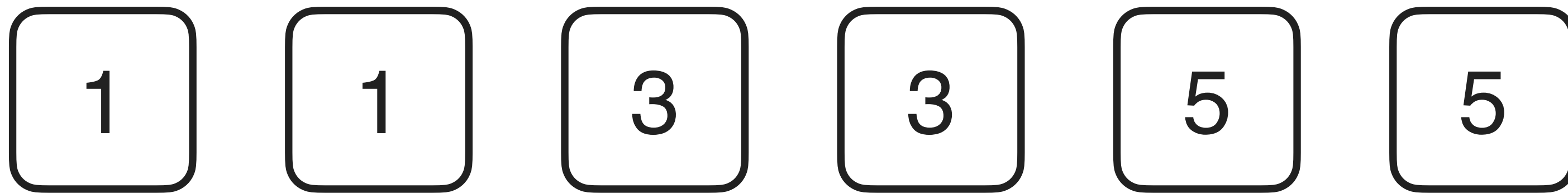
$$m = l + (r - l) / 2 = 0 + (1 - 0) / 2 = 0$$

搜索空间没有变化

lower 的问题

if(arr[mid] < target) l = mid

if(arr[mid] >= target) r = mid - 1



$$m = l + (r - l) / 2 = 0 + (1 - 0) / 2 = 0$$

搜索空间没有变化

这个问题只有在 l 和 r 相临时存在

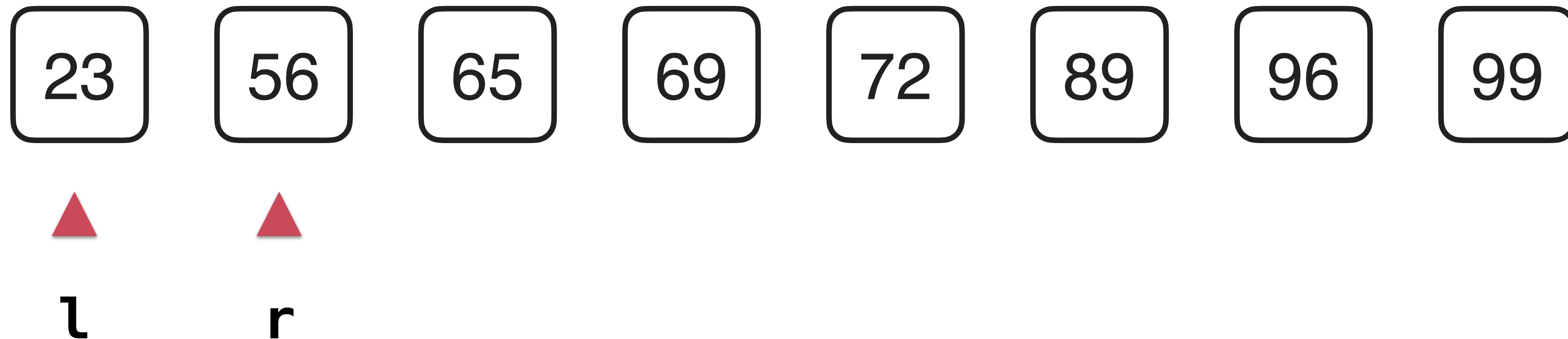
解决方案: $m = l + (r - l + 1) / 2$

upper

查找大于 target 的最小值 搜索范围 arr[l, r] r=arr.length

if(arr[mid] > target) r = mid

if(arr[mid] <= target) l = mid + 1



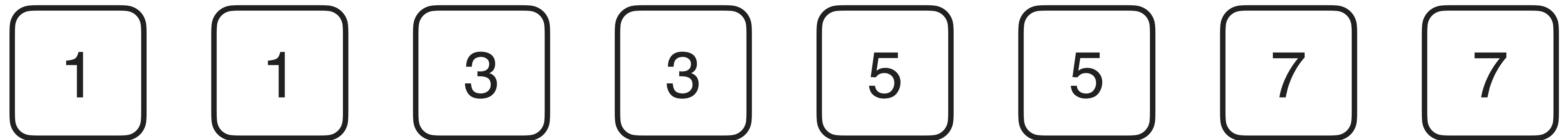
实践：解决 lower 的问题

lower_floor 和 upper_floor

lower_floor

查找 5 如果数组中存在元素，返回最小索引

查找 4

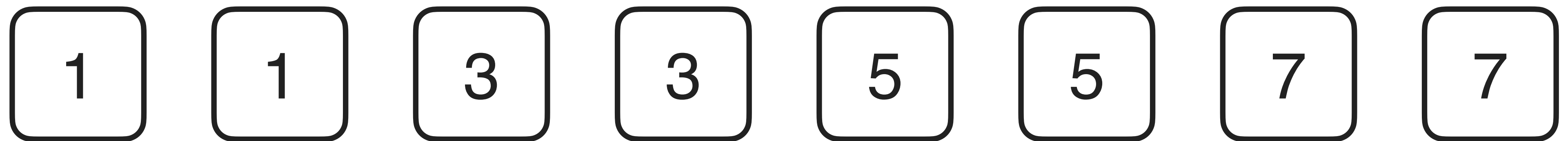


r

lower_floor

查找 5 如果数组中存在元素，返回最小索引

查找 4 如果数组中不存在元素，返回 lower

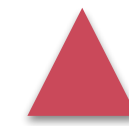
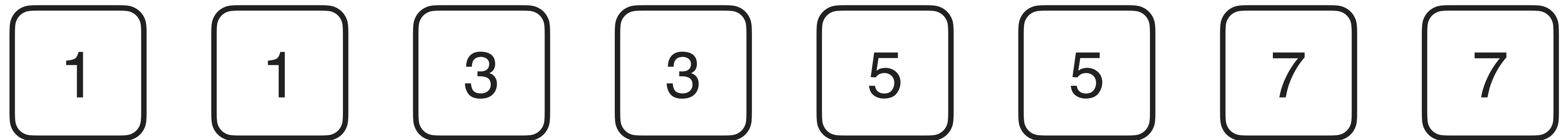


r

upper_floor

查找 5 如果数组中存在元素，返回最大索引

查找 4

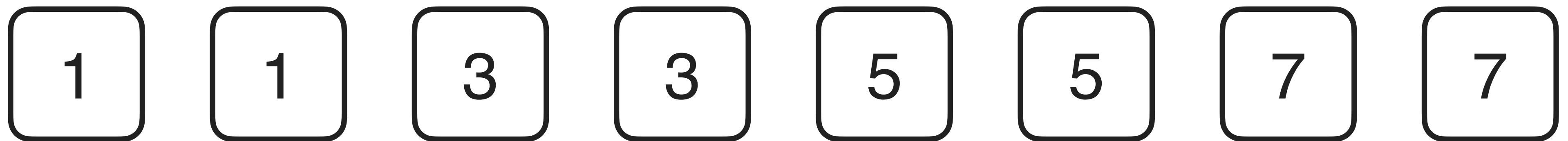


r

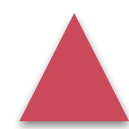
upper_floor

查找 5 如果数组中存在元素，返回最大索引

查找 4 如果数组中不存在元素，返回 lower



\leq target 的最大索引



r

作业解析： lower_floor 和 upper_floor

二分查找法模板

二分查找法模板

二分查找

upper

lower

upper_ceil

upper_floor

lower_ceil

lower_floor

二分查找法模板

```
int l =  , r =  ;
```

// 在 $data[l, r]$ 中寻找解

```
while(l < r){
```

```
    int mid = 上取整? 下取整?
```

```
    if(data[mid].compareTo(target)   0)
```

```
         
```

```
    else
```

```
         
```

```
}
```

```
return l;
```

作业：用 \geq target 的最小值的思路；
实现二分查找法

作业解析：用 \geq target 的最小值的思路；
实现二分查找法

其他

欢迎大家关注我的个人公众号：是不是很酷



算法与数据结构体系课程

liuyubobobo