

0704二分查找練習

Given an array of integers `nums` which is **sorted** in ascending order, and an integer `target`, write a function to search `target` in `nums`. If `target` exists, then return its index. Otherwise, return `-1`.

提示1:

```
Input: nums = [-1,0,3,5,9,12], target = 9
```

```
Output: 4
```

```
Explanation: 9 exists in nums and its index is 4
```

提示2:

```
Input: nums = [-1,0,3,5,9,12], target = 2
```

```
Output: -1
```

```
Explanation: 2 does not exist in nums so return -1
```

Note:

1. `nums`中的所有元素是不重複的。
2. `n`將在`[1, 10000]`之間
3. `nums`的每個元素都將在`[-9999, 9999]`之間

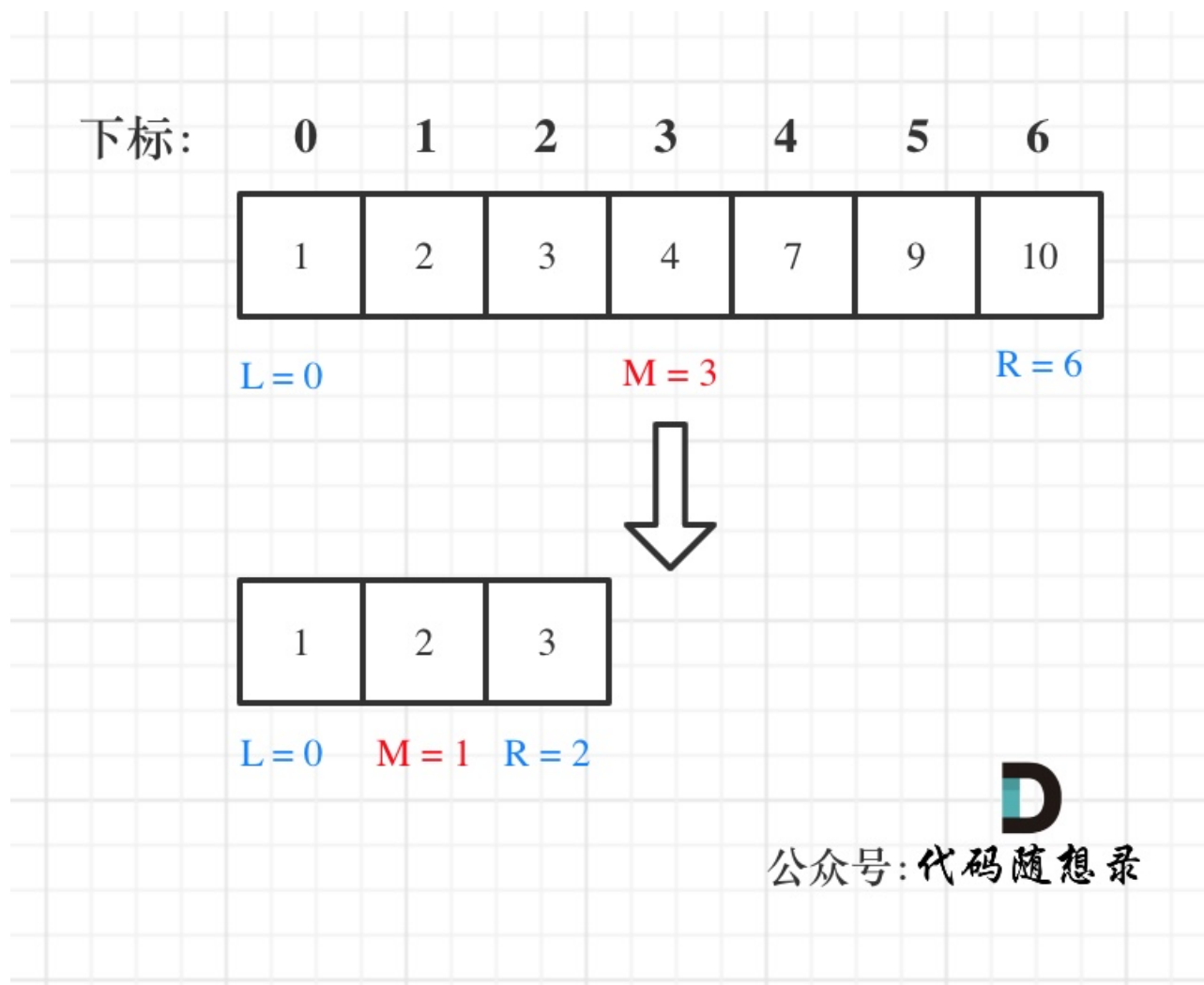
想法

二分法最重要的就是對區間的定義，區間的定義就是不變量，在二分查找過程中飽時不變量，就是while尋找中每一次邊界的處裡都要堅持根據區間的定義來操作，這就是循環不變量規則。

方法1:

定義`target`在左閉右閉的區間(**[left, right]**)，需要注意以下兩點:

- `while(left <= right)`要使用 `<=`，因為`left == right`是有意義的。
- `if (nums[middle] > target)` `right` 要賦值為 `middle - 1`，因為當前這個`nums[middle]`一定不是`target`，那接下來要查找的左區間結束下標位置就是 `middle - 1`。

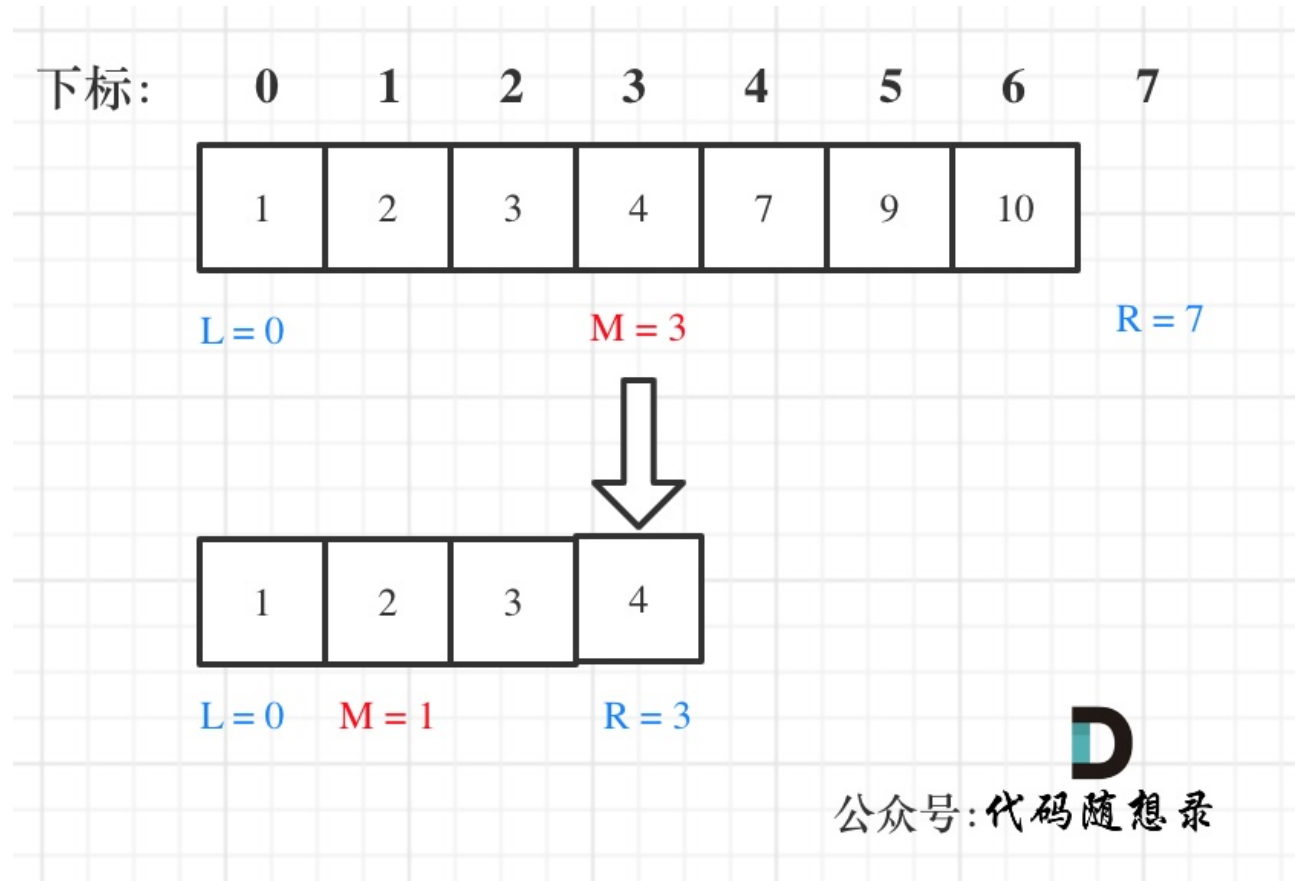


方法2:

定義target在左閉右開的區間(**left, right**)，需要注意以下兩點:

- while(left < right)要使用 <，因為left == right是沒有意義的。
- if (nums[middle] > target) right 要賦值為 middle，因為當前這個nums[middle]不等於target，去左區間繼續尋找，而尋找區間是左閉右開，所以right更新為middle，即下一個查詢區間不會去比較

nums[middle]。



以下為自寫程式碼:

法1:

```
//法1 左閉右閉
int search(int* nums, int numsSize, int target){
    int left = 0;
    int right = numsSize - 1;
    while(left <= right){
        int middle = (left + (right - left)/2);
        if (nums[middle] > target){
            right = middle - 1; //即target在左區間
        }
        else if (nums[middle] < target){
            left = middle + 1;
        }
        else{
            return middle;
        }
    }
    return -1; //未找到target
}
```

法2:

```
//法2 左閉右開
int search(int* nums, int numsSize, int target){
    int left = 0;
    int right = numsSize;
    while(left < right){
        int middle = (left + (right - left)/2);
        if (nums[middle] > target){
            right = middle; //即target在左區間
        }
        else if (nums[middle] < target){
            left = middle + 1;
        }
        else{
            return middle;
        }
    }
    return -1; //未找到target
}
```