Bài 5.3: Thuật toán binary search

- ✓ Các đặc điểm thuật toán
- ✓ Mã giả và triển khai
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Các đặc điểm thuật toán

- ✓Thuật toán tìm kiếm nhị phân có độ phức tạp O(logn).
- ✓Là thuật toán tìm kiếm nhanh nhất trong số các thuật toán tìm kiếm.
- ✓Yêu cầu tiên quyết của thuật toán là các phần tử của tập mẫu phải được sắp xếp trước khi tiến hành tìm kiếm.
- ✓Khi tìm kiếm ta cần biết chỉ số phần tử trái, phải của đoạn cần tìm kiếm.
- ✓Gọi left, right là các chỉ số trái cùng, phải cùng của đoạn cần xét. Giả sử ta cần tìm x và gọi chỉ số phần tử giữa đoạn đang xét là mid.
- ✓Nếu left <= right:
  - ✓ Nếu x == giá trị phần tử tại vị trí mid, tìm thấy x, return chỉ số mid;
  - ✓ Nếu x < phần tử tại vị trí mid, tiến hành tìm kiếm nhị phân nửa bên trái vị trí mid;
  - ✓ Nếu x > phần tử tại vị trí mid, tiến hành tìm kiếm nhị phân nửa bên phải vị trí mid;
- ✓Nếu left > right: không tìm thấy x, return -1;

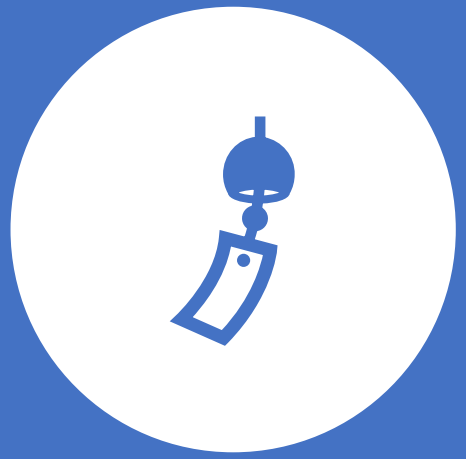
Mã giả của thuật toán

✓Sau đây là mã giả của thuật toán tìm kiếm nhị phân:

```
// thuật toán tìm kiếm nhị phân
// arr: mảng chứa các giá trị để tìm kiếm
// left: chỉ số phần tử trái cùng của mảng
// right: chỉ số phần tử phải cùng của đoạn
// x: giá trị cần tìm
int binarySearch (arr[], left, right, x) {
    if (left <= right) {
        mid = left + (right - left) / 2;
        if (arr[mid] == x) { // tìm thấy x trong mảng
            return mid;
        }
        if (arr[mid] < x) { // tìm phía bên phải arr[mid]
            return binarySearch(arr, mid + 1, right, x);
        } else { // tìm phía trái arr[mid]
            return binarySearch(arr, left, mid - 1, x);
        }
    }
    return -1; // không tìm thấy x trong mảng
}
```

Triển khai của thuật toán

```
3 references
static int BinarySearch(int[] arr, int x, int left, int right)
{
    if(left <= right)
    {
        int mid = left + (right - left) / 2;
        if(arr[mid] == x) // tìm thấy x rồi!
        {
            return mid; // trả về vị trí tìm thấy đầu tiên
        }
        if(arr[mid] < x) // tìm kiếm nửa bên phải mid
        {
            return BinarySearch(arr, x, mid + 1, right);
        } else // tìm kiếm nửa bên trái mid
        {
            return BinarySearch(arr, x, left, mid - 1);
        }
    }
    return -1; // không tìm thấy
}
```



Nội dung tiếp theo

Tìm hiểu lớp Array