ĐẠI NAM
UNIVERSITY

Phần trình bày của:

ĐẬU HẢI PHONG

Giảng viên

Đại Nam, ngày 01 tháng 12 năm 2022

LƯU Ý

KHÔNG NÓI CHUYỆN RIÊNG

KHÔNG SỬ DỤNG ĐIỆN THOẠI

KHÔNG NGỦ GẬT

GHI CHÉP ĐẦY ĐỦ

ĐẠI NAM
UNIVERSITY

CẤU TRÚC DỮ LIỆU & GIẢI THUẬT

CHƯƠNG 5

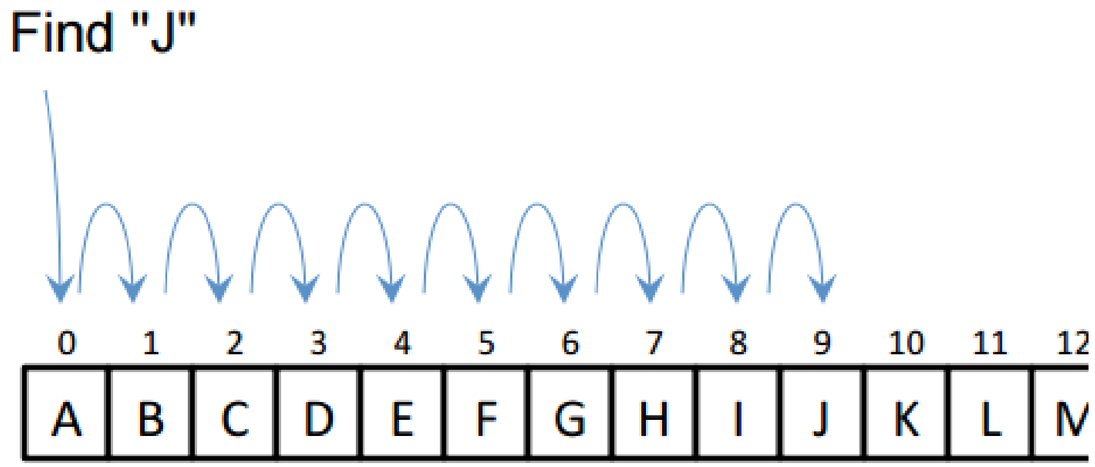
THUẬT TOÁN TÌM KIẾM

NỘI DUNG

- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Cây nhị phân tìm kiếm
- Cây AVL
- Bài tập

Thuật toán tìm kiếm tuần tự

- Tìm kiếm tuyến tính là gì?
  - Tìm một phần tử trong một danh sách
  - Duyệt lần lượt từng phần tử trong danh sách đến khi:
    - Tìm thấy hoặc hết danh sách
  - Danh sách không cần sắp xếp



Thuật toán tìm kiếm tuần tự

- Ý tưởng:
  - Duyệt từ đầu đến cuối mảng
    - So sánh từng phần tử với phần tử cần tìm kiếm (x)
    - Nếu phần tử so sánh bằng x thì trả về vị trí của mảng
  - Nếu duyệt hết mảng mà không tìm thấy thì trả về -1
- Độ phức tạp của thuật toán:
  - Tốt nhất: O(1)
  - Xấu nhất: O(n)

Thuật toán tìm kiếm tuần tự

```
3  int linearSearch(int arr[], int n, int x)
4  {
5      //Lặp từng phần tử của mảng và kiểm tra.
6      for(int i = 0; i < n; i++)
7          if (arr[i] == x)
8              return i;
9      return -1;// Trả về -1 nếu đã duyệt hết mà ko tìm thấy.
10 }
```

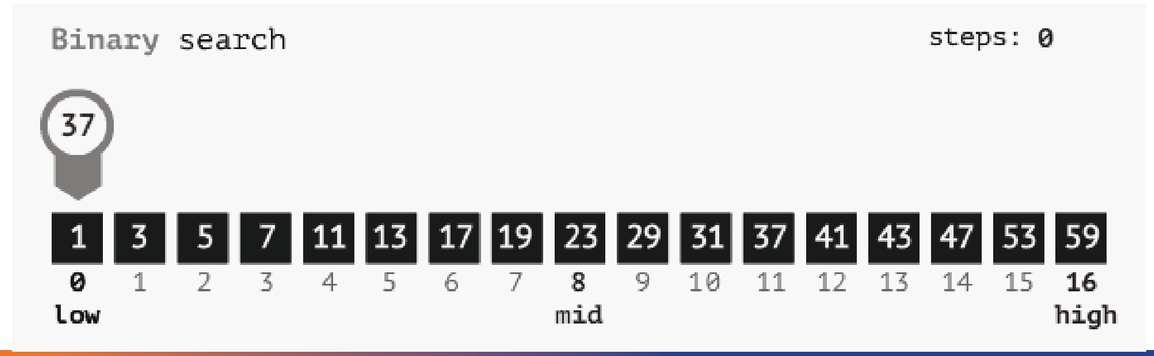
```
11 int main()
12 {
13     int arr[] = {1, 5, 12, -10, 5, 11}; //Khởi tạo mảng
14     int x = -10; //Phần tử cần tìm kiếm
15     // Số phần có trong mảng.
16     int n = sizeof(arr) / sizeof(arr[0]); //Cho biết kích thước mảng n = 6
17     int result = linearSearch(arr, n, x);
18     if (result == -1)
19         cout << "Khong tim thay " << x <<" trong mang";
20     else
21         cout << "Vi tri: " << result;
22 }
```

Tìm kiếm nhị phân

- Tìm kiếm nhị phân là gì?
  - Là tìm kiếm một phần tử trên một danh sách đã được sắp xếp
  - Tìm phần tử giữa để danh sách thành 2 phần và tìm trên phần *(giả sử sắp xếp tăng dần)*
    - Nếu nhỏ hơn phần tử giữa thì tìm bên trái
    - Nếu lớn hơn phần tử giữa thì tìm bên phải

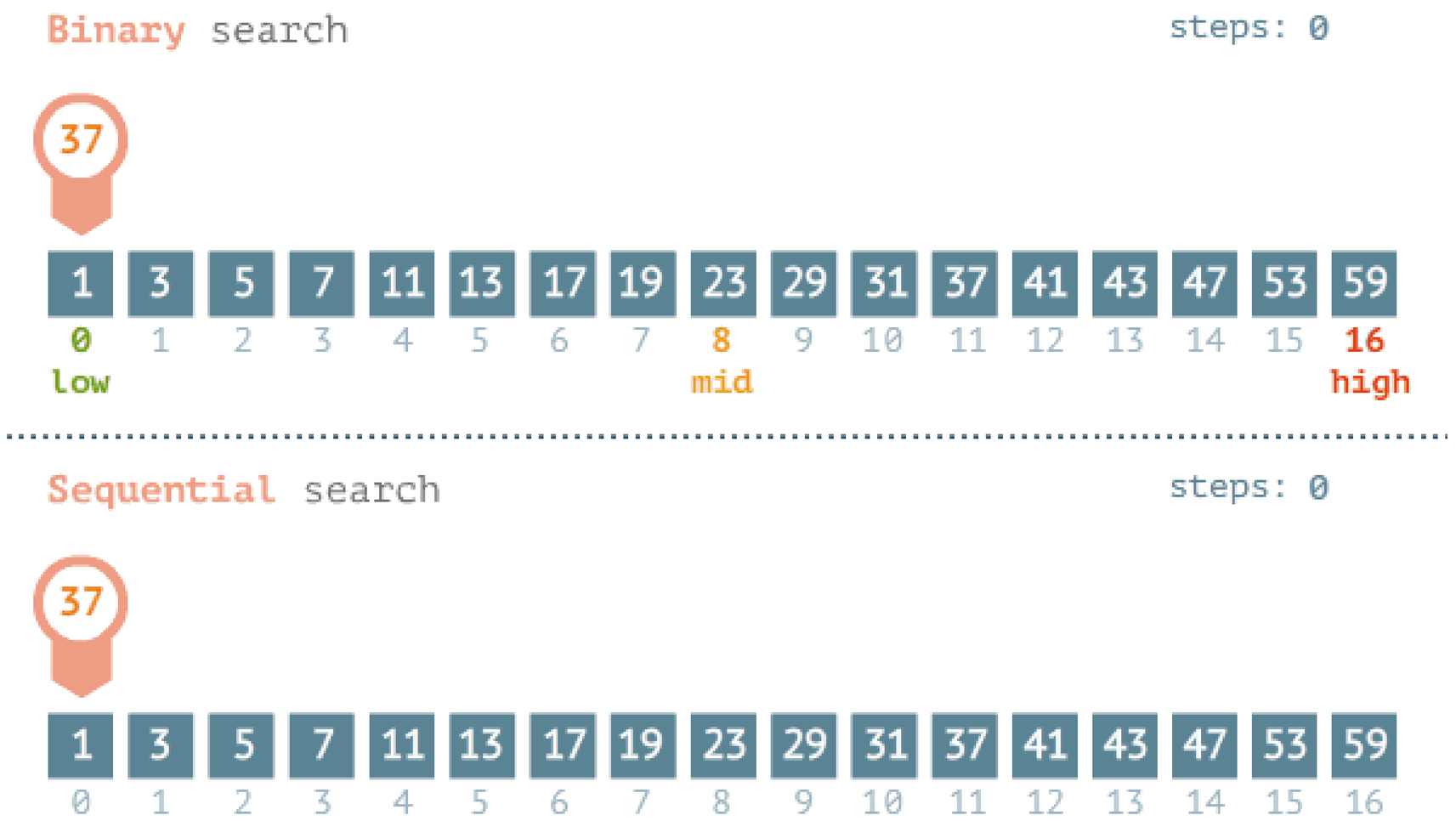
Tìm kiếm nhị phân

- Ý tưởng: *(giả sử sắp xếp tăng dần)*
  - Xét đoạn mảng a[left…right], cần tìm phần tử x
  - mid = (left + right)/2
  - Nếu x = a[mid] thì trả về mid và thoát
  - Nếu x < a[mid] thì tìm trên a[mid+1,right]
  - Nếu x > a[mid] thì tìm trên a[left,mid-1]



Tìm kiếm nhị phân

- Tìm kiếm nhị phân và Tìm kiếm tuyến tính



Tìm kiếm nhị phân

```
4  // Hàm tìm kiếm nhị phân sử dụng giải thuật đệ quy
5  int binarySearch(int arr[], int l, int r, int x)
6  {
7    if (r >= l)
8    {
9      int mid = l + (r - l) / 2; // Tương đương (l+r)/2 nhưng ưu điểm tránh tràn số khi l,r lớn
10
11     // Nếu arr[mid] = x, trả về chỉ số và kết thúc.
12     if (arr[mid] == x)
13       return mid;
14
15     // Nếu arr[mid] > x, thực hiện tìm kiếm nửa trái của mảng
16     if (arr[mid] > x)
17       return binarySearch(arr, l, mid - 1, x);
18
19     // Nếu arr[mid] < x, thực hiện tìm kiếm nửa phải của mảng
20     return binarySearch(arr, mid + 1, r, x);
21   }
22   return -1;// Nếu không tìm thấy
23 }
```

Hỏi - Đáp

Bài tập

- Bài 1.

ĐẠI NAM
UNIVERSITY

Cảm ơn đã lắng nghe!

ĐẬU HẢI PHONG
Giảng viên
dauhaiphong@dainam.edu.vn
0912441435