

BÀI TẬP CHỦ ĐỀ 2: TÌM KIẾM

Phần trắc nghiệm (chọn 1 câu trả lời đúng nhất)

Câu 1. Phát biểu nào sau đây là đúng ?

- A. Tìm kiếm tuần tự và lính canh đều có thời gian chạy là $O(n)$
- B. Tìm kiếm nhị phân luôn luôn chạy nhanh hơn tìm kiếm tuần tự trong mọi trường hợp.
- C. Tìm kiếm lính canh luôn có thời gian chạy nhanh hơn tìm kiếm tuần tự trong mọi trường hợp.
- D. Không có câu nào đúng

Câu 2. Phần tử nào được dùng làm lính canh trong tìm kiếm lính canh?

- A. Phần tử đầu tiên của mảng
- B. Phần tử cuối cùng của mảng
- C. Phần tử cần tìm kiếm (khóa)
- D. Phần tử ở giữa mảng

Câu 3. Cho mảng arr chưa được sắp thứ tự. Đoạn code nào dưới đây là tìm kiếm tuần tự?

```
A. int unorderedLinearSearch(int arr[], int size, int data)
{
    int index;
    for(int i = 0; i < size; i++)
    {
        if(arr[i] == data)
        {
            index = i;
            break;
        }
    }
    return index;
}
```

```
B. int unorderedLinearSearch(int arr[], int size, int data)
{
    int index;
    for(int i = 0; i < size; i++)
    {
        if(arr[i] == data)
        {
            break;
        }
    }
    return index;
}
```

```
}
```

```
C. int unorderedLinearSearch(int arr[], int size, int data)
{
    int index;
    for(int i = 0; i <= size; i++)
    {
        if(arr[i] == data)
        {
            index = i;
            break;
        }
    }
    return index;
}
```

D. Tất cả các câu trên đều sai

Câu 4. Cho mảng arr đã được sắp thứ tự tăng dần. Đoạn code nào dưới đây là tìm kiếm tuần tự?

```
A. int LinearSearch(int arr[], int size, int data)
{
    int index = -1;
    int i = 0;
    while(size > 0)
    {
        if(data[i] == key)
        {
            index = i;
        }
        if(data[i] > key)
        {
            index = i;
            break;
        }
        i++;
    }
    return index;
}
```

```
B. int LinearSearch(int arr[], int size, int data)
{
    int index = -1;
    int i = 0;
    while(size > 0)
    {
        if(data[i] == key)
        {
            index = i;
        }
        if(data[i] > key)
        {
            break;
        }
        i++;
    }
    return index;
}
```

```
}
```

```
C. int LinearSearch(int arr[], int size, int data)
{
    int index = -1;
    int i = 0;
    while(size > 0)
    {
        if(data[i] == key)
        {
            index = i;
        }
        if(data[i] > key)
        {
            index = i;
        }
        i++;
    }
    return index;
}
```

D. Tất cả các câu trên đều sai

Câu 5: Trường hợp tốt nhất, tìm kiếm tuần tự chạy trong:

- A. $O(\log_2 n)$
- B. $O(n \log_2 n)$
- C. $O(1)$
- D. $O(n)$

Câu 6: Trường hợp xấu nhất, tìm kiếm tuần tự chạy trong:

- A. $O(\log_2 n)$
- B. $O(n \log_2 n)$
- C. $O(1)$
- D. $O(n)$

Câu 7: Phát biểu nào sau đây cho biết nhược điểm của thuật toán tìm kiếm tuần tự?

- A. Tốn nhiều không gian lưu trữ hơn tìm kiếm nhị phân
- B. Thời gian chạy thuật toán lâu hơn
- C. Khó cài đặt hơn thuật toán tìm kiếm nhị phân
- D. Tất cả các câu trên

Câu 8: Thuật toán tìm kiếm nào sau đây đòi hỏi mảng đã cho phải được sắp thứ tự sẵn.

- A. Tìm kiếm tuần tự
- B. Tìm kiếm lính canh
- C. Tìm nhị phân
- D. Tất cả các thuật toán trên

Câu 9: Phát biểu nào dưới đây KHÔNG phải là một hạn chế/nhược điểm của tìm kiếm nhị phân?

- A. Chỉ áp dụng tìm kiếm trên mảng đã được sắp thứ tự
- B. Tốn thời gian sắp xếp mảng nếu mảng chưa được sắp thứ tự
- C. Phần tử ở giữa mảng phải được truy xuất trực tiếp dễ dàng
- D. Tìm kiếm nhị phân không hiệu quả với mảng có số lượng phần tử lớn hơn 1000

Câu 10: Trường hợp xấu nhất, tìm kiếm nhị phân có thời gian chạy:

- A. Tương đương thuật toán tìm kiếm tuần tự
- B. Lâu hơn thuật toán tìm kiếm tuần tự
- C. Nhanh hơn thuật toán tìm kiếm tuần tự
- D. Có lúc nhanh hơn, có lúc chậm hơn thuật toán tìm kiếm tuần tự

Phần tự luận

Câu 1. Hãy cài đặt thuật toán tìm kiếm tuần tự bằng đệ quy.

Câu 2. Hãy cài đặt thuật toán in ra màn hình các số giống nhau trong một mảng có n số nguyên .

Câu 3. Hãy cài đặt thuật toán in ra màn hình số có tần số xuất hiện cao nhất trong mảng có n số nguyên.

Câu 4. Xét mảng các số nguyên có nội dung như sau:

$$A = \langle -9, -9, -5, -2, 0, 3, 7, 7, 10, 15 \rangle$$

- a. Tính số lần so sánh để tìm ra phần tử $x = -9$ bằng phương pháp tìm tuần tự và tìm nhị phân. Nhận xét và so sánh 2 phương pháp tìm nêu trên trong trường hợp này.
- b. Trong trường hợp tìm nhị phân, phần tử nào sẽ được tìm thấy (thứ 1 hay 2)?

Câu 5. Hãy xây dựng thuật toán tìm phần tử trung vị (median) của một dãy số a_1, a_2, \dots, a_n . Cho biết độ phức tạp của thuật toán này?

- o Gợi ý: Trung vị của một dãy A gồm n phần tử (đã sắp thứ tự) được tính theo công thức:

$$Me = A\left[\frac{n-1}{2}\right] \text{ nếu } n \text{ lẻ, } Me = \frac{A\left[\frac{n}{2}-1\right] + A\left[\frac{n}{2}\right]}{2} \text{ nếu } n \text{ chẵn.}$$

Câu 6: Hãy liệt kê các phần tử được so sánh với khoá k trong quá trình tìm nhị phân trên mảng $A = \langle 0, 2, 4, 7, 8, 14, 28, 45 \rangle$ trong các trường hợp sau:

- a. $k = 2$
- b. $k = 8$
- c. $k = 15$

Câu 7. Cho mảng A gồm n số nguyên. Hãy xây dựng thuật toán tìm một phần tử trong mảng A với 2 bước:

- o Bước 1: Sắp xếp mảng A bằng một thuật toán sắp xếp đã học (tùy chọn)
- o Bước 2: Tìm kiếm tuần tự trên mảng A đã sắp
- a. Cho biết thuật toán này có ưu điểm gì so với thuật toán tìm kiếm tuần tự trên mảng chưa sắp thứ tự? (Gợi ý: xét trường hợp tìm thấy và trường hợp không tìm thấy)
- b. Thuật toán có chạy nhanh hơn thuật toán tìm kiếm nhị phân hay không?

c. Hãy chạy thử nghiệm thuật toán với mảng $A = \langle 19, 50, 10, 49, 80, 7, 2, 5 \rangle$ và phần tử cần tìm x như sau:

1. $x = 2$
2. $x = 80$
3. $x = 79$
4. $x = 1$
5. $x = 9$

Câu 8. Hãy cài đặt thuật toán tìm kiếm nhị phân bằng vòng lặp (không dùng đệ quy).