Thuật toán tìm kiếm nhị phân – Binary Search

1. Mô tả code:
   1. Input:

- Đầu vào của bài toán là yêu cầu nhập vào một số n nguyên và dãy n số nguyên được lưu vào 1 mảng 1 chiều

- Nhập 1 số nguyên x để tìm kiếm vị trí

* 1. Mô tả code:

- Sau khi nhập xong mảng a thì sẽ cần sắp xếp lại mảng a theo chiều tăng dần

- Cho 2 biến l = 0, r = n mục đích là tìm kiếm trong đoạn từ 0 đến n và cho 1 biến pos = -1 để lưu lại vị trí nếu tìm thấy

- Sau đó dùng vòng lặp while(l <= r)

- Trong vòng while tạo 1 biến m = (l + r) / 2 // tìm vị trí ở giữa mảng a

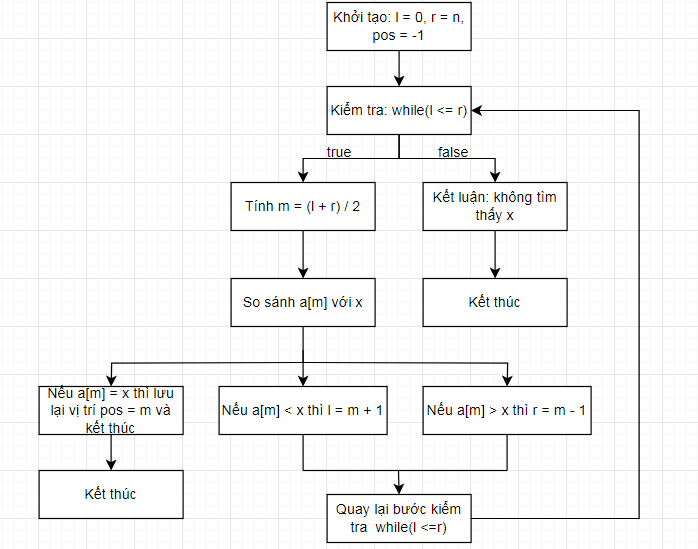
- Nếu a[m] == x // nếu giá trị phần tử thứ m của mảng a = x thì lưu m vào biến pos và thoát vòng lặp

- Nếu a[m] < x thì l = m+1 // nếu giá trị ở vị trí m bé hơn giá trị cần tìm thì cho biến l = m + 1 tức là x chỉ nằm trong khoảng giá trị > a[m] nên chỉ cần tìm trong khoảng từ m+1 ->n

- Nếu a[m] < x thì ngược lại giá trị cần tìm chỉ nằm trong khoản từ 0 đến m–1.

- Lặp lại các bước trên cho đến khi tìm được x hoặc l > r (không tìm thấy x)

* 1. Output: đầu ra của bài toán là vị trí của số nguyên x cần tìm
  2. Lược đồ thuật toán:



* 1. Demo

#Testcase 1:

input:

Nhap n: 5

1 3 5 6 7

Nhap gia tri can tim: 3

output:

Vi tri gia tri can tim la: 2

#Testcase 2:

input:

Nhap n: 5

1 6 3 2 8

Nhap gia tri can tim: 3

output:

Vi tri gia tri can tim la: 3

#Testcase 3:

input:

Nhap n: 5

1 6 3 2 8

Nhap gia tri can tim: 2

output:

Vi tri gia tri can tim la: 2

#Testcase 4:

input:

Nhap n: 5

1 6 3 2 8

Nhap gia tri can tim: 4

output:

"Khong tim thay!"

#Testcase 5:

input:

Nhap n: 5

1 6 3 2 8

Nhap gia tri can tim: 10

output:

"Khong tim thay!"