

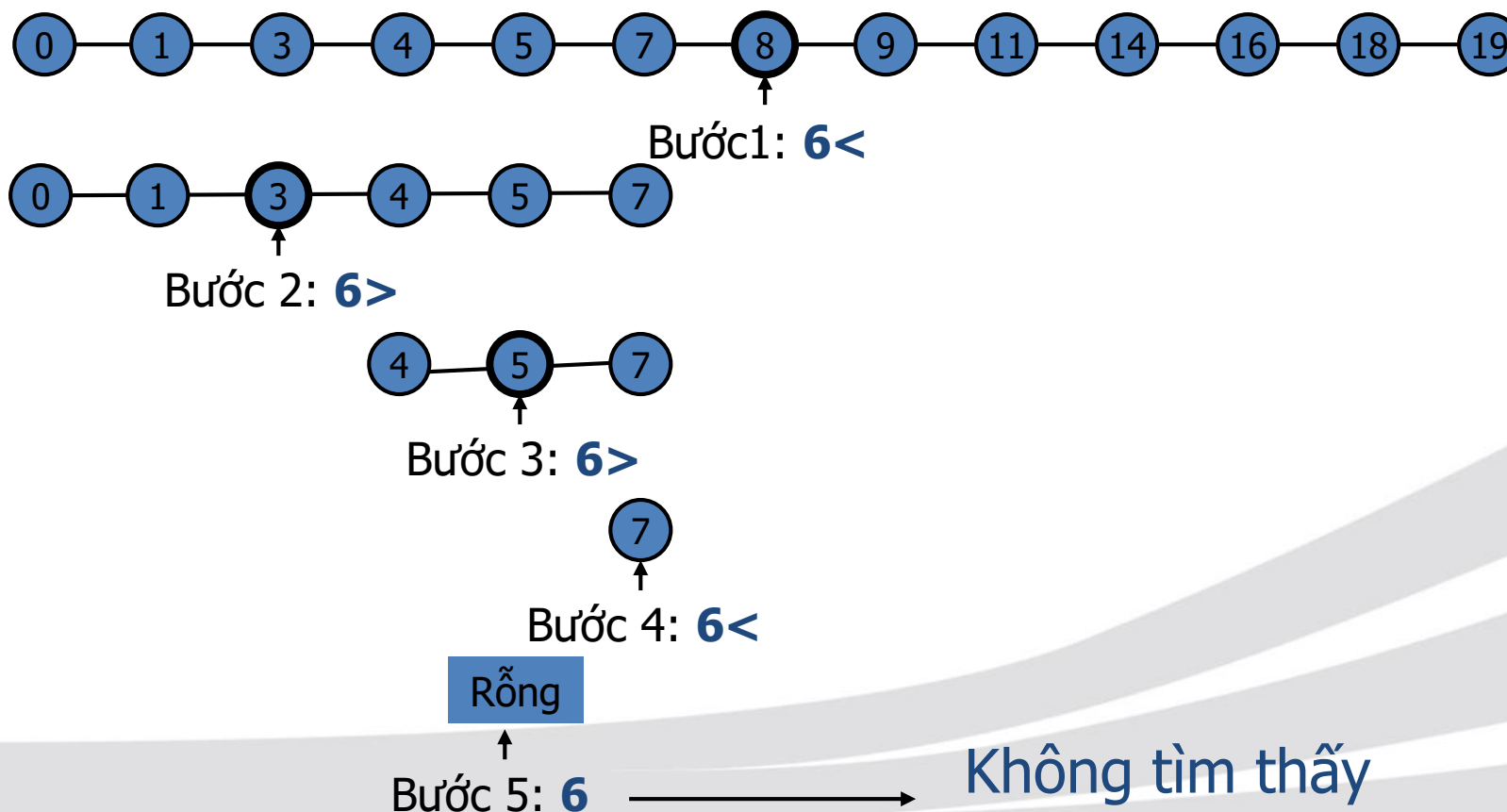
## Thuật toán Tìm kiếm nhị phân trên mảng

- ❑ Các phần tử của **S** được lưu trữ trong mảng và được sắp xếp theo thứ tự tăng dần (giảm dần) của giá trị khóa (key).
- ◆ Thuật toán tìm kiếm nhị phân được thiết kế dựa trên chiến lược loại trừ
- ◆ Thuật toán: So sánh khóa **k** với khóa của phần tử ở giữa dãy.
  - Nếu trùng thì thông báo tìm thấy và dừng
  - Nếu  $k >$  thì gọi đệ qui tìm trên nửa cuối dãy
  - Nếu  $k <$  thì gọi đệ qui tìm trên nửa đầu dãy
  - Quá trình tìm nếu phải tìm trong dãy rỗng thì dừng lại và thông báo không tìm thấy

# Ví dụ 1



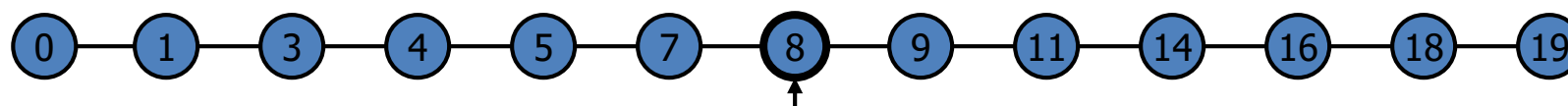
- Cho dãy dưới đây. Tìm phần tử  $k=6$



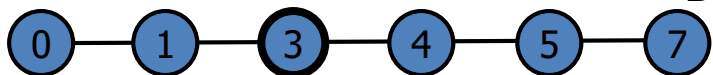
# Ví dụ 2



- Cho dãy dưới đây. Tìm phần tử  $k=5$



Bước 1:  $5 <$



Bước 2:  $5 >$



Bước 3:  $5 =$



Tìm thấy

**Algorithm** BinarySearch(S, k, n);

  i = 1;

  j = n;

**while**(i ≤ j){

    mid = (i+j) / 2;

**if** (k == S[mid].Key  
       return mid;

**else**

**if** (k < S[mid].Key)  
        j = mid - 1;

**else**

        i = mid+1;

  }

**return** -1;

# Thời gian chạy



- ❑ Sau mỗi lần tìm kiếm, thì dãy cần tìm lại được chia đôi và ta chỉ phải tìm trên một nửa
- ❑ Trong trường hợp xấu nhất là không tìm thấy phần tử có khóa k. Và như vậy ta phải thực hiện chia đôi liên tiếp đến khi được dãy rỗng. Số lần thực hiện chia đôi là:  $\log_2 n$
- ❑ Vậy thời gian chạy là  $O(\log_2 n)$

# Hết