Thuật toán tìm kiếm nhị phân

Bởi:

Khoa CNTT ĐHSP KT Hưng Yên

Bài toán :

Cho mảng gồm n phần tử đã được sắp xếp tăng dần và một phần tử x. Tìm xem x có trong mảng hay không? Nếu có x trong mảng thì trả ra kết quả là 1, nếu không trả ra kết quả là 0.

Dùng thuật toán tìm kiếm nhị phân,

Phân tích thuật toán :

Số x cho trước

+ Hoặc là bằng phần tử nằm ở vị trí giữa mảng

+ Hoặc là nằm ở nửa bên trái (x < phần tử ở giữa mảng)

+ Hoặc là nằm ở nửa bên phải (x > phần tử ở giữa mảng)

Mô tả thuật toán:

Input: mảng a[1..n]

Output: + 1 nếu x thuộc a

+ 0 nếu x không thuộc a

Từ nhận xét đó ta có giải thuật sau:

```
Tknp (a, x, Đầu, Cuối)
If (Đầu > Cuối)
 return  0 ; {dãy trống}
Else
{   Giữa = (Đầu + cuối) / 2;
   If (x == a[Giữa])
   Return 1;
else
        if (x > a[Giữa])
                Tknp(a, x, Giữa + 1, Cuối) ;
        else
                Tknp(a, x, Đầu, Giữa - 1) ;
}
```

Đánh giá độ phức tạp thời gian của thuật toán

Trường hợp tốt nhất: Tương ứng với sự tìm được y trong lần so sánh đầu tiên, tức là x= a[giữa] (x nằm ở vị trí giữa mảng)

=> Ta có : $T_{tốt}$ (n) = O(1)

Trường hợp xấu nhất: Độ phức tạp là O(log n)

Thật vậy, Nếu gọi T(n) là độ phức tạp của thuật toán, thì sau khi kiểm tra điều kiện (x == a[giữa]) và sai thì gọi đệ qui thuật toán này với dữ liệu giảm nửa, nên thỏa mãn công thức truy hồi :

Thuật toán tìm kiếm nhị phân

$T(n) = 1 + T(n/2)$ với $n>=2$ và $T(1) = 0$