BÀI 25 THỰC HÀNH XÁC ĐỊNH ĐỘ PHỨC TẠP THỜI GIAN THUẬT TOÁN

SAU BÀI HỌC NÀY EM SẼ:

- Thực hành xác định độ phức tạp (O-lớn) của hàm thời gian.

Biết cách phân tích, đánh giá độ phức tạp thuật toán là kĩ năng quan trọng của người thiết kế thuật toán và chương trình. Các quy tắc đơn giản tính độ phức tạp thời gian mang lại cho em điều gì khi đánh giá thuật toán?

Nhiệm vụ 1

Xác định độ phức tạp thời gian tính toán của thuật toán tìm kiếm tuần tự được thể hiện bằng chương trình sau:

```
1 def LinearSearch(A,K):
2     for i in range(len(A)):
3         if A[i] == K:
4             return i
5     return -1
```

Hướng dẫn:

Bước 1. Phân tích thời gian tính toán của thuật toán.

Gọi n là kích thước của mảng đầu vào, T(n) là thời gian thực hiện thuật toán.

– Đối với mã lệnh trên, chương trình sẽ thực hiện duyệt mảng và với mỗi bước lặp sẽ kiểm tra phần tử thứ i có bằng với phần tử cần tìm kiếm không (dòng 3). Nếu bằng, thì chương trình sẽ trả về chỉ số của phần tử tìm thấy và kết thúc (dòng 4). Như vậy, chương trình có thể kết thúc khi chưa duyệt hết mảng (trường hợp đã tìm thấy phần tử) và tối đa là duyệt hết mảng. Như vậy, trong trường hợp tồi nhất vòng lặp ở dòng 2 sẽ thực hiện n bước lặp, mỗi bước lặp sẽ thực hiện lệnh so sánh ở dòng 3 tốn 1 đơn vị thời gian.

– Lệnh trả về sẽ được thực hiện duy nhất 1 lần ở dòng 4 (trường hợp tìm thấy phần tử trong mảng) hoặc dòng 5 (trường hợp không tìm thấy phần tử trong mảng) mất 1 đơn vị thời gian.

Do đó, tổng số phép tính cơ bản của chương trình trong trường hợp tồi nhất là

$$T(n) = n+1.$$

Bước 2. Xác định độ phức tạp O-lớn của thuật toán

$$T(n) = n + 1 = O(n+1) = O(\max(n, 1)) = O(n).$$

Vậy thuật toán tìm kiếm tuần tự có độ phức tạp tuyến tính.

Nhiệm vụ 2

Xác định độ phức tạp của thuật toán sắp xếp chọn được thể hiện bằng chương trình sau:

```
1 def SelectionSort(A):
2     n = len(A)
3     for i in range(n-1):
4         iMin = i
5         for j in range(i+1,n):
6             if A[j] < A[iMin]:
7                 iMin = j
8         A[i],A[iMin] = A[iMin],A[i]
```

Hướng dẫn:

Ý tưởng của thuật toán sắp xếp chọn là tại mỗi bước thứ i của vòng lặp sẽ tìm chính xác phần tử tại vị trí thứ i, tức là sẽ tìm phần tử nhỏ nhất trong dãy từ A[i], A[i + 1], …, A[n – 1] và đổi chỗ phần tử nhỏ nhất này với A[i].

Gọi n là kích thước của mảng A, T(n) là thời gian chạy của thuật toán. Thời gian chạy của thuật toán được phân tích như sau:

– Lệnh gán ở dòng 2 tốn 1 đơn vị thời gian.

– Vòng lặp tại dòng 3 biến i sẽ chạy từ 0 đến n – 2, vậy vòng lặp này có n – 1 bước lặp.

– Tại mỗi bước lặp của lệnh **for** tại dòng 3 chương trình sẽ thực hiện các lệnh sau:

+ Lệnh gán tại dòng 4 tốn 1 đơn vị thời gian

+ Vòng lặp **for** tại lệnh 5, biến j sẽ chạy từ i + 1 đến n – 1, nên vòng lặp này có n – i – 1 bước lặp.

+ Với mỗi bước lặp tại dòng 5 chương trình sẽ thực hiện:

1 lệnh so sánh tại dòng 6 tốn 1 đơn vị thời gian và một lệnh gán tại dòng 7 tốn 1 đơn vị thời gian (nếu điều kiện thoả mãn).

Như vậy mỗi bước lặp tại dòng 5 sẽ tốn tối đa 2 đơn vị thời gian.

+ 1 lệnh đổi chỗ tại dòng 8 tốn 3 đơn vị thời gian.

Tổng hợp lại ta thấy thời gian chạy chương trình trên là:

$$T(n) = 1 + \sum_{i=0}^{n-2} \left(1 + 2(n - i - 1) + 3\right)$$

$$T(n) = 1 + 4(n - 1) + 2\sum_{i=0}^{n-2} (n - i - 1)$$

$$T(n) = 1 + 4(n - 1) + 2\sum_{k=1}^{n-1} k$$

$$T(n) = 1 + 4(n - 1) + n(n - 1)$$

$T(n) = n^2 + 3n - 3$

Xác định độ phức tạp O-lớn của thuật toán:

$T(n) = O(\max(n^2, 3n, -3)) = O(n^2)$

Vậy thuật toán sắp xếp chọn có độ phức tạp thời gian bình phương.

LUYỆN TẬP

1. Xác định độ phức tạp của thuật toán sắp xếp nổi bọt sau:

```
1 def BubbleSort(A):
2     n = len(A)
3     for i in range(n-1):
4         for j in range(n-1-i):
5             if A[j] > A[j+1]:
6                 A[j],A[j+1] = A[j+1],A[j]
```

2. Cho biết hàm sau sẽ trả về giá trị là bao nhiêu? Xác định độ phức tạp thời gian O-lớn của chương trình.

```
1 def Mystery(n):
2     r = 0
3     for i in range(n-1):
4         for j in range(i+1,n):
5             for k in range(1,j):
6                 r = r+1
7     return r
```


VẬN DỤNG

1. Giả sử rằng mỗi phép tính đơn được thực hiện trong micro giây (1 μs = một phần triệu giây). Hãy xác định giá trị lớn nhất của n trong các thuật toán tìm kiếm tuần tự, sắp xếp chèn và sắp xếp chọn nếu thời gian thực thi các thuật toán là 1 giây, 1 phút và 1 giờ?

2. Hãy cho biết hàm sau thực hiện công việc gì? Xác định độ phức tạp thời gian của thuật toán.

```
1 def func(A):
2     n = len(A)
3     for i in range(n-1):
4         for j in range(i+1,n):
5             if A[i] > A[j]:
6                 A[i],A[j] = A[j],A[i]
```