Chương 1

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Bài Tập Lý Thuyết

Câu 1: Đếm số phép toán gán, phép so sánh được thực thi và xác định độ phức tạp trong đoạn code sau:

for (i = 0; i < n; i++)

for (j = 0; j < m; j++)

if (a[ i ][ j ] = = x) return 1;

return -1;

Trả lời:

Phép gán:

Với n = 0 có số phép gán là 1;

Với n = 1 có số phép gán là (n + 1) + 1;

Với n = 2 có số phép gán là 2(n + 1) + 1;

…

Với n = n có số phép gán là n(n + 1) + 1.

Phép so sánh:

Với n = 0 có số phép so sánh là 1;

Với n = 1 có số phép so sánh là 2n + 1;

Với n = 2 có số phép so sánh là 2(n + 1);

…

Với n = n có số phép so sánh là n(2n + 1).

Độ phức tạp của thuật toán thuộc lớp O(n2).

Câu 2: Đếm số phép toán gán, phép so sánh được thực thi và xác định độ phức tạp trong đoạn code sau:

sum = 0; for (i = 0; i < n ; i++)

for (j = 0; j < i ; j++)

sum++;

Trả lời:

Phép gán:

Với n = 0 có số phép gán là 1;

Với n = 1 có số phép gán là (2n+1)+1;

Với n = 2 có số phép gán là 2(2n+1)+1;

…

Với n = n có số phép gán là n(2n+1)+1.

Phép so sánh:

Với n = 0 có số phép so sánh là 1;

Với n = 1 có số phép so sánh là (n+1)+1;

Với n = 2 có số phép so sánh là 2(n+1)+1;

…

Với n = n có số phép so sánh là n(n+1)+1.

Độ phức tạp của thuật toán thuộc lớp O(n2).

Câu 3: Đánh giá độ phức tạp của đoạn code sau:

for (i = 0; i < n; i++)

sum1+=i;

for (i = 0; i < n\*n; i++)

sum2+=i;

Trả lời: Đoạn code trên 2 vòng lặp for ko lồng nhau nên độ phức tạp của thuật toán thuộc lớp O(n).

Câu 4: Đánh giá độ phức tạp của hàm tính giai thừa sau:

int GT (int n)

{

if (n == 1)

return 1;

return n\*GT(n-1);

}

Trả lời: Đoạn code trên thuộc bài toán đệ quy nên độ phức tạp của thuật toán trên thuộc lớp O(n2).

Câu 5: Đánh giá độ phức tạp của hàm tính dãy FIBONACCI sau:

int Fibo (int n)

{

if (n <=1)

return n;

return Fibo(n-1) + Fibo(n-2);

}

Trả lời: Đoạn code trên thuộc bài toán đệ quy nên độ phức tạp của thuật toán trên thuộc lớp O(n2).