Bài 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;

giải:

Phép gán:

Vòng lặp trong:

m = 0: có 1 gán

m = 1: có 2 gán

m= 2: có 3 gán

…

m = m: có m + 1 gán

Vòng lặp ngoài:

n = 0: có (m + 1) \* 0 + 1 gán

n = 1: có (m + 1) \* 1 + 1 gán

…

n = n: có (m + 1) \* n + 1 gán

* T(m, n) = (m + 1) \* n + 1

Nếu m = n => (n + 1) \* n + 1 ⬄ O(n2)

Phép so sánh

Vòng lặp trong:

m = 0: có 1 so sánh

m = 1: có 3 so sánh

…

m = m: có m \* 2 + 1 gán

Vòng lặp ngoài:

n = 0: có (m \* 2 + 1) \* 0 + 1 gán

n = 1: có (m \* 2 + 1) \* 1 + 1 gán

…

n = n: có (m \* 2 + 1) \* n + 1 gán

* T(m, n) = (m \* 2 + 1) \* n + 1

Nếu m = n => (n \* 2 + 1) \* n + 1 ⬄ O(n2)

Độ phức tạp của thuật toán là : O(n2).