**Câu 1:**

-Đếm phép gán:

Với n = 0 thì ta có 1 phép gán.

Với n = 1 thì ta có 1 + (m + 1) phép gán.

Với n = 2 thì ta có 1 + 2(m + 1) phép gán.

Với n = 3 thì ta có 1 + 3(m + 1) phép gán.

->Vậy m = n thì ta sẽ có 1 + n(n + 1) phép gán ~ O(n2).

-Đếm phép so sánh:

Với n = 0 thì ta có 1 phép so sánh.

Với n = 1 thì ta có 1 + 2m phép so sánh.

Với n = 2 thì ta có 1 + 2\*2m phép so sánh.

Với n = 3 thì ta có 1 + 2\*3m phép so sánh.

->Vậy m = n thì ta sẽ có 1 + 2\*n\*n so sánh ~ O(n2).

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

**Câu 2:**

-Đếm phép gán:

Với n = 0 thì ta có 2 phép gán.

Với n = 1 thì ta có 1 + (2 \* 1 + 1) phép gán.

Với n = 2 thì ta có 1 + 2(2 \* 2 + 1) phép gán.

Với n = 3 thì ta có 1 +3(2 \* 3 + 1) phép gán.

->Vậy ta sẽ có 1 + n(2 \* n + 1) = 2n2 + n + 1 phép gán ~ O(n2).

-Đếm phép so sánh:

Với n = 0 thì ta có 1 phép so sánh.

Với n = 1 thì ta có 1 + (1 + 1) phép so sánh.

Với n = 2 thì ta có 1 + 2(2 + 1) phép so sánh.

Với n = 3 thì ta có 1 + 3(3 + 1) phép so sánh.

->Vậy ta sẽ có 1 + n(n + 1) = n2 + n + 1 so sánh ~ O(n2).

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

**Câu 3:**

Ta thấy 2 dòng lặp for nằm riêng biệt -> Độ phức tạp của giải thuật là O(n).

**Câu 4:**

Ta thấy cấu trúc gọi hàm đệ quy -> Độ phức tạp của thuật toán là O(n).

**Câu 5:**

Ta thấy cấu trúc gọi hàm đệ quy -> Độ phức tạp của thuật toán là O(n).