Độ phức tập của thuật toán :

kí hiệu O(n)

\*ghi nhớ :

O(c.f(n)) = O(f(n))

O(c) = O(1)

\* Các quy tắc đánh giá độ phức tạp thuật toán

1.quy tắc cộng

g/s: p1 có thời gian thực hiện là T1(n) = O(f(n))

p2 có thời gian thực hiện là T2(n) = O(g(n))

khi đó thời gian thực hiện của đoạn p1 nối tiếp p2 là max(O(f(n)), O(g(n)))

2.quy tắc nhân

g/s: p1 có thời gian thực hiện là T1(n) = O(f(n))

p2 có thời gian thực hiện là T2(n) = O(g(n))

khi đó thời gian thực hiện của đoạn p1, p2 lồng vào nhau là

O(f(n)) . O(g(n)) = O(f(n).g(n))

\* Cách đánh giá chi tiết cho từng chương trình

- thời gian thực hiện cho các câu lệnh đơn gồm I/O, gán, goto --> O(1)

- thời gian thực hiện cho các khối lệnh tuần tự nhau thì sử dụng QT cộng

- thời gian thực hiện cho cấu trúc if

gọi thời gian thực hiện cho các câu lệnh sau **then** là O(f(n)) và thời gian thực hiện cho các câu lệnh sau **else** là O(g(n)) và thời gian thực hiện kiểm tra điều kiện là O(1) ==> QT cộng

- thời gian thực hiện cho cấu trúc lặp **for**

gọi thời gian thực hiện cho các câu lệnh sau **do** (tức là các câu lệnh sẽ làm việc cho mỗi lần lặp) O(f(n)), gọi thời gian thực hiện vòng lặp O(g(n)) ==> QT nhân O(f(n) . g(n))