*Bài tâp 2.1: Mô tả quá trình sắp xếp dãy số bằng INSERTIONSORT*

40 70 20 60 90 10 50 30

j = 0 i=1 2 3 4 5 6 7

// gọi a[0] la phần tử được xem là dnah sách đã dược xếp thứ tự

\* gán x = a[i] và j = i - 1

-xét i = 1 vào \*: so sánh a[i] = 70 > a[j] = 40, i++

-xét i = 2 vào \*: so sánh a[i] = 20 < a[j] = 70

+1: gán a[j + 1] = a[j], j-- ( j = 0 )

so sánh x = 20 < a[j] = 40, quay lại +1 và dừng lại j < 0 or x > a[j]

gán a[j+1] = x, i++

=> dãy sau 2 lần sắp xếp:

20 40 70 60 90 10 50 30

0 1 2 i=3 4 5 6 7

-xét i = 3 vào \*: so sánh a[i] = 60 < a[j] = 70

+1: gán a[j + 1] = a[j], j-- ( j = 1 )

so sánh x = 60 > a[j] = 40 , không quay lại +1

gán a[j + 1] = x, i++

=> dãy sau sắp xếp lần 3:

20 40 60 70 90 10 50 30

0 1 2 3 i = 4 5 6 7

-xét i = 4 vào \*: so sánh a[i] = 90 > a[j] = 70, i++

-xét i = 5 vào \*: so sánh a[i] = 10 < a[j] = 90

+1: gán a[j + 1] = a[j], j-- ( j = 3 )

so sánh x = 10 < a[j] = 70 , quay lại +1 và dừng khi j < 0 or x > a[j]

gán a[j + 1] = x, i++

=> dãy sau sắp xếp lần 4:

10 20 40 60 70 90 50 30

0 1 2 3 4 5 i=6 7

-xét i = 6 vào \*: so sánh a[i] = 50 > a[j] = 90

+1: gán a[j + 1] = a[j], j-- ( j = 4 )

so sánh x = 50 < a[j] = 70 , quay lại +1 và dừng khi j < 0 or x > a[j]

gán a[j + 1] = x, i++

=> dãy sau sắp xếp lần 5:

10 20 40 50 60 70 90 30

0 1 2 3 4 5 6 i=7

-xét i = 7 vào \*: so sánh a[i] = 30 < a[j] = 90

+1: gán a[j + 1] = a[j], j-- ( j = 5 )

so sánh x = 30 < a[j] = 70 , quay lại +1 và dừng khi j < 0 or x > a[j]

gán a[j + 1] = x, i++

=> dãy sau sắp xếp lần 6:

10 20 30 40 50 60 70 90

0 1 2 3 4 5 6 7

Dừng chương trình khi i > số phần tử - 1

Ta dược sắp xếp dãy thwo thứ tựu tăng dần bằng *INSERTIONSORT*

10 20 30 40 50 60 70 90

Độ phức tạp thuật toán: **O(n^2)**