BÀI TẬP LÀM THÊM CHƯƠNG 3

Bài 2 – Câu 2.1: Cho danh sách các phần tử được lưu trữ trong 1 danh sách đặc có các phần tử sau: 40, 70, 20, 60, 90, 10, 50, 30. Yêu cầu dùng phương pháp sắp xếp thứ tự InsertionSort, mô tả từng bước quá trình (không lập trình). Tín độ phức tạp của quá trình sắp xếp trên.

Giải

void InsertionSort(int a[], int n)

{

int x;

for (int i = 1; i < n; i++)

{

x = a[i];

j = i – 1;

while (j >= 0 && a[j] > x)

{

a[j + 1] = a[j];

j--;

}

a[j + 1] = x;

}

}

Chạy tay:

n = 8

i = 1

x = a[i] = a[1] = 70

j = 0 >= 0 && a[j] = a[0] = 40 < x -> sai điều kiện -> dừng while

a[j + 1] = a[1] = x = 70 //40 70 20 60 90 10 50 30

i = 2

x = a[i] = a[2] = 20

j = 1 >= 0 && a[j] = a[1] = 70 > x

a[j + 1] = a[2] = a[j] = a[1] = 70 //40 70 70 60 90 10 50 30

j--

j = 0 >= 0 && a[j] = a[0] = 40 > x

a[j + 1] = a[1] = a[j] = a[0] = 40 //40 40 70 60 90 10 50 30

j--

j = -1 < 0 -> sai điều kiện -> dừng while

a[j + 1] = a[0] = x = 20 //20 40 70 60 90 10 50 30

i = 3

x = a[i] = 60

j = 2 >= 0 && a[j] = a[2] = 70 > x

a[j + 1] = a[3] = a[j] = a[2] = 70 //20 40 70 70 90 10 50 30

j--

j = 1 >= 0 && a[j] = a[1] = 40 < x -> sai điều kiện -> dừng while

a[j + 1] = a[2] = x = 60 //20 40 60 70 90 10 50 30

i = 4

x = a[i] = a[4] = 90

j = 3 >= 0 && a[j] = a[3] = 70 < x -> sai điều kiện -> dừng while

a[j + 1] = a[4] = x //20 40 60 70 90 10 50 30

i = 5

x = a[i] = a[5] = 10

j = 4 >= 0 && a[j] = a[4] = 90 > x

a[j + 1] = a[5] = a[j] = a[4] = 90 //20 40 60 70 90 90 50 30

j--

j = 3 >= 0 && a[j] = a[3] = 70 > x

a[j + 1] = a[4] = a[j] = a[3] = 70 //20 40 60 70 70 90 50 30

j--

j = 2 >= 0 && a[j] = a[2] = 60 > x

a[j + 1] = a[3] = a[j] = a[2] = 60 //20 40 60 60 70 90 50 30

j--

j = 1 >= 0 && a[j] = a[1] = 40 > x

a[j + 1] = a[2] = a[j] = a[1] = 40 //20 40 40 60 70 90 50 30

j--

j = 0 >= 0 && a[j] = a[0] = 20

a[j + 1] = a[1] = a[j] = a[0] = 20 //20 20 40 60 70 90 50 30

j--

j = -1 < 0 -> sai điều kiện -> dừng while

a[j + 1] = a[0] = x = 10 //10 20 40 60 70 90 50 30

i = 6

x = a[i] = a[6] = 50

j = 5 >= 0 && a[j] = a[5] = 90 > x

a[j + 1] = a[6] = a[j] = a[5] = 90 //10 20 40 60 70 90 90 30

j--

j = 4 >= 0 && a[j] = a[4] = 70 > x

a[j + 1] = a[5] = a[j] = a[4] = 70 //10 20 40 60 70 70 90 30

j--

j = 3 >= 0 && a[j] = a[3] = 60 > x

a[j + 1] = a[4] = a[j] = a[3] = 60 //10 20 40 60 60 70 90 30

j--

j = 2 >= 0 && a[j] = a[2] = 40 < x -> sai điều kiện -> dừng while

a[j + 1] = a[3] = x //10 20 40 50 60 70 90 30

i = 7

x = a[i] = a[7] = 30

j = 6 >= 0 && a[j] = a[6] = 90 > x

a[j + 1] = a[6] = a[j] = a[5] = 90 //10 20 40 50 60 70 90 90

j--

j = 5 >= 0 && a[j] = a[5] = 70 > x

a[j + 1] = a[6] = a[j] = a[5] = 70 //10 20 40 50 60 70 70 90

j--

j = 4 >= 0 && a[j] = a[4] = 60 > x

a[j + 1] = a[5] = a[j] = a[4] = 60 //10 20 40 50 60 60 70 90

j--

j = 3 >= && a[j] = a[3] = 50 > x

a[j + 1] = a[4] = a[j] = a[3] = 50 //10 20 40 50 50 60 70 90

j--

j = 2 >= 0 && a[j] = a[2] = 40 > x

a[j + 1] = a[3] = a[j] = a[2] = 50 //10 20 40 40 50 60 70 90

j--

j = 1 >= 0 && a[j] = a[1] = 20 < x -> sai điều kiện -> đừn while

a[j + 1] = a[2] = x //10 20 30 40 50 60 70 90

i = 8 <= n -> sai điều kiện dừng for i

Sau khi thực hiện hàm ta thu được mảng sắp xếp tăng là :

10 20 30 40 50 60 70 90

# Độ phức tạp của giải thuật:

Ta biết:

* Vòng lặp for có độ phức tạp thuộc lớp O(n)
* Vòng lặp while có độ phức tạp thuộc lớp O(n)

Mà 2 vòng lặp này lồng vào nhau nên suy ra độ phức tạp của giải thuật trên là thuộc lớp O(n2).