Cấu trúc dữ liệu và giải thuật

**Bài 2:** Một danh sách các phần tử được lưu trữ trong một danh sách đặc, có các phần tử sau: 40, 70, 20, 60, 90, 10, 50, 30. Yêu cầu:

2.1 Dùng phương pháp xếp thứ tự InsertionSort, mô tả từng bước quá trình xếp thứ tự dãy số trên (không lập trình). Tín độ phức tạp của quá trình xếp thứ tự danh sách trên.

Ta có danh sách :

A={ 40, 70, 20, 60, 90, 10, 50, 30}

A[]=a[0] a[1] a[2] a[3] a[4] a[5] a[6] a[7]

Ta chọn n là số lượng phần tử có trong danh sách ;

Và i là phần tử đầu trong danh sách khi này i = 1; và i< n;

Và j là phần tử thứ i-1 => j = i-1; và j >=0;

Và x sẽ được gán bằng a[i];

Ta có i = 1; => j =0; x = a[1] =70;

Xét j>=0 và x < a[j]

Ta có (j=0) nên không thỏa điều kiện trên

Ta gán giá trị của j+1 bằng x;

Ta tăng i lên 1;

Khi này ta được dãy: a 40 70 20 60 90 10 50 30

Ta có i = 2; => j =1; x = a[2] =20;

Xét j>=0 và x < a[j]

Ta có (j=1) nên thỏa điều kiện trên nên ta có:

Giá trị của a[2] = giá trị của a[1]; => a[2] = 70;

Và giảm j xuống 1 bậc khi này j =0 thỏa điều kiện nên ta có

Giá trị của a[1] = giá trị của a[0]; => a[1]=40;

Và giảm j xuống 1 bậc khi này j không thỏa điều kiện nên ta có:

Ta gán giá trị của a[0] bằng x;

Ta tăng i lên 1;

Khi này ta được dãy: a 20 40 70 60 90 10 50 30

Ta có i = 3; => j =2; x = a[3] =60;

Xét j>=0 và x < a[j]

Ta có (j=2) nên thỏa điều kiện trên nên ta có:

Giá trị của a[3] = giá trị của a[2]; => a[3] = 70;

Và giảm j xuống 1 bậc khi này j =1 không thỏa điều kiện nên ta có

Ta gán giá trị của a[2] bằng x;

Ta tăng i lên 1;

Khi này ta được dãy: a 20 40 60 70 90 10 50 30

Ta có i = 4; => j =3; x = a[4] =90;

Xét j>=0 và x < a[j]

Ta có (j=3) không thỏa điều kiện trên nên ta có:

Ta gán giá trị của a[4] bằng x;

Ta tăng i lên 1;

Khi này ta được dãy: a 20 40 60 70 90 10 50 30

Ta có i = 5; => j =4; x = a[5] =10;

Xét j>=0 và x < a[j]

Ta có (j=4) thỏa điều kiện trên nên ta có:

Giá trị của a[5] = giá trị của a[4]; => a[5] = 90;

Và giảm j xuống 1 bậc khi này j =3 thỏa điều kiện nên ta có

Giá trị của a[4] = giá trị của a[3]; => a[4]=70;

Và giảm j xuống 1 bậc khi này j =2 thỏa điều kiện nên ta có

Giá trị của a[3] = giá trị của a[2]; => a[3]=60;

Và giảm j xuống 1 bậc khi này j =1 thỏa điều kiện nên ta có

Giá trị của a[2] = giá trị của a[1]; => a[2]=40;

Và giảm j xuống 1 bậc khi này j =0 thỏa điều kiện nên ta có

Giá trị của a[1] = giá trị của a[0]; => a[1]=20;

Và giảm j xuống 1 bậc khi này j không thỏa điều kiện nên ta có:

Ta gán giá trị của a[0] bằng x;

Ta tăng i lên 1;

Khi này ta được dãy: a 10 20 40 60 70 90 50 30

Ta có i = 6; => j =5; x = a[6] =50;

Xét j>=0 và x < a[j]

Ta có (j=5) thỏa điều kiện trên nên ta có:

Giá trị của a[6] = giá trị của a[5]; => a[6] = 90;

Và giảm j xuống 1 bậc khi này j =4 thỏa điều kiện nên ta có

Giá trị của a[5] = giá trị của a[4]; => a[5]=70;

Và giảm j xuống 1 bậc khi này j =3 thỏa điều kiện nên ta có

Giá trị của a[4] = giá trị của a[3]; => a[4]=60;

Và giảm j xuống 1 bậc khi này j =2 không thỏa điều kiện nên ta có:

Ta gán giá trị của a[3] bằng x;

Ta tăng i lên 1;

Khi này ta được dãy: a 10 20 40 50 60 70 90 30

Ta có i = 7; => j =6; x = a[7] =30;

Xét j>=0 và x < a[j]

Ta có (j=6) thỏa điều kiện trên nên ta có:

Giá trị của a[7] = giá trị của a[6]; => a[7] = 90;

Và giảm j xuống 1 bậc khi này j =5 thỏa điều kiện nên ta có

Giá trị của a[6] = giá trị của a[5]; => a[6]=70;

Và giảm j xuống 1 bậc khi này j =4 thỏa điều kiện nên ta có

Giá trị của a[5] = giá trị của a[4]; => a[5]=60;

Và giảm j xuống 1 bậc khi này j =3 thỏa điều kiện nên ta có

Giá trị của a[4] = giá trị của a[3]; => a[4]=50;

Và giảm j xuống 1 bậc khi này j =3 thỏa điều kiện nên ta có

Giá trị của a[3] = giá trị của a[2]; => a[3]=40;

Và giảm j xuống 1 bậc khi này j =2 không thỏa điều kiện nên ta có:

Ta gán giá trị của a[2] bằng x;

Ta tăng i lên 1;

Khi này ta được dãy: a 10 20 30 40 50 60 70 90

Khi tăng i lên 1 không thỏa điều kiện i< n nên ta kết thúc chương trình và thu được kết quả:

Dãy: a 10 20 30 40 50 60 70 90

Tính độ phức tạp của thuật toán:

void InsertionSort(int []a, int n)

{

int x, i, j;

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

{

x = a[i];

j = i - 1;

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

{

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

j--;

}

a[j+1] = x;

}

}

Ta thấy vòng lặp for có 3 lệnh gán : x = a[i]; j = i - 1; a[j+1] = x;

Ta thấy vòng lặp while có 5 lệnh: 0 <= j; &&; x < a[j]; a[j + 1] = a[j]; j--;

Nên ta có biểu thức T(n) = 3 + 5\*i

Mà I chạy từ 1 đến n nên ta có

T(n) = 3+5\*1 + 3+5\*2 + 3+5\*3 +….+ 3+5\*n

T(n) = 3\*n + 5\*(1+2+3+…+n)

T(n) = 3n + 5n(n+1)/2

T(n) = (5/2)\*n^2 + (11/2)\*n

* Độ phức tạp của thuật toán là O(n^2)