Tên: Huỳnh Thị Tuyết Ngọc

Lớp IT01

Nhóm 7

Bài 2:Dãy số: 40 70 20 60 90 10 50 30

2.1:InsertionSort

Giả sử a[0] đã có thứ tự tăng dần

* Xét a[1] = 70 > a[0] = 40 => chèn a[1] vào bên phải của a[0].

KQ: 40 70 20 60 90 10 50 30

* Xét a[2] = 20 < a[0] = 40 => chèn a[2] vào trước của a[0].

KQ: 20 40 70 60 90 10 50 30

* Xét a[3] = 60 > a[1] = 40 => chèn a[3] vào bên phải của a[1].

KQ: 20 40 60 70 90 10 50 30

* Xét a[4] = 90 > a[3] = 70 => chèn a[4] vào bên phải của a[0].

KQ: 20 40 60 70 90 10 50 30

* Xét a[5] = 10 < a[0] = 20 => chèn a[5] vào trước a[0].

KQ: 10 20 40 60 70 90 50 30

* Xét a[6] = 50 > a[2] = 40 => chèn a[6] vào bên phải a[2].

KQ: 10 20 40 50 60 70 90 30

* Xét a[7] = 30 >a[1] = 20 => chèn a[7] vào bên phải a[1].

KQ: 10 20 30 40 50 60 70 90.

Độ phức tạp của thuật toán trên thuộc lớp O(n2)

2.2 SelectionSort

40, 70, 20, 60, 90, 10,50, 30

i = 0 < i = n -1. Tìm min\_pos trong dãy từ i=0 đến i=n-1.

min\_pos = 5. Hoán đổi a[5] với a[0].

Ta được: 10, 70, 20,60, 90, 40, 50, 30.

i++;

i = 1 < i = n -1. Tìm min\_pos trong dãy từ i=1 đến i=n-1.

min\_pos = 2. Hoán đổi a[2] với a[1].

Ta được: 10, 20, 70,60, 90, 40, 50, 30.

i++;

i = 2 < i = n -1. Tìm min\_pos trong dãy từ i=2 đến i=n-1.

min\_pos = 7. Hoán đổi a[7] với a[2].

Ta được: 10, 20, 30,60, 90, 40, 50, 70.

i++;

Tương tự, ta thực hiện cho đến khi dãy sắp xếp hoàn toàn.

KQ: 10, 20, 30, 40, 50, 60, 70, 90.

Độ phức tạp của thuật toán trên thuộc lớp O(n2)

2.3 InterchangeSort

Dãy số: 40 70 20 60 90 10 50 30

So sánh a[0] = 40 < a[1]= 70 không hoán đổi

So sánh a[0] = 40 > a[2]= 20 hoán đổi a[0] với a[1].

Kq: 20 70 40 60 90 10 50 30

So sánh a[0] = 20 < a[3]= 60 không hoán đổi

So sánh a[0] = 20 < a[4]= 90 không hoán đổi

Tương tự, ta làm đến khi nhận được kết quả

KQ: 10, 20, 30, 40, 50, 60, 70, 90.

Độ phức tạp của thuật toán trên thuộc lớp O(n2)

2.4BubbleSort

Dãy số: 40 70 20 60 90 10 50 30

So sánh a[7]= 30 < a[6] = 50. Hoán đổi a[6] và a[5].

Kq: 40 70 20 60 90 10 30 50

So sánh a[6]= 30 > a[5] = 10. Không hoán đổi

So sánh a[5]= 10 < a[4] = 90. Hoán đổi a[5] và a[4].

Kq: 40 70 20 60 10 90 30 50

Tương tự, ta làm đến khi nhận được kết quả

KQ: 10, 20, 30, 40, 50, 60, 70, 90.

Độ phức tạp của thuật toán trên thuộc lớp O(n2)

2.5 QuickSort

Dãy số: 40 70 20 60 90 10 50 30

Left = 0, right = 7

X= a[(left + right)/2] = a[3]= 60.

a[i] = a[0]= 40 < 60, i++

a[i]=a[1]= 70 > 60, vi phạm, dừng i

a[j] = a[7] = 30 < 60, vi phạm, dừng j

hoán đổi a[1] và a[7]

kq: 40 30 20 60 90 10 50 70

i++, j—

a[2] = 20 < 60, i++;

a[i] = a[3] = 60 < 60, vi phạm, dừng i

a[j] = a[6] = 50 < 60, vi phạm, dừng j

hoán đổi a[i] = a[3] và a[j] = a[6].

kq: 40 30 20 50 90 10 60 70

i++, j--;

a[i] = a[4] = 90 > 60 vi phạm, dừng i

a[j] = a[5] = 10 < 60, vi phạm, dừng j

hoán đổi a[4] và a[5]

kq: 40 30 20 50 10 90 60 70

i++,j--,

i > j, dừng

tiếp tục chia đoạn cho đến khi dãy được sắp xếp hoàn toàn.

Độ phức tạp của thuật toán trên thuộc lớp O(n log n)

2.6 HeapSort

Dãy số: 40 70 20 60 90 10 50 30

Chia dãy thành 2 đoạn: a[0] đến a[n/2-1]=a[3] và a[4] đến a[7]

Tại vị trí i=3, so sánh a[3] vs a[2\*3+ 1]= a[7] và a[2\*3 + 2] = 8

Giá trị lớn nhất của a[3], a[7] và a[8] là a[3]=60, không hoán đổi

Giảm i xuống i= 2, so sánh a[2] vs a[5], a[6]. Giá trị lớn nhất là a[6]. Hoán đổi a[2] vs a[6], ta được:

40 70 50 60 90 10 20 30

Giảm i xuống, i=1, so sánh a[1] vs a[3], a[4]. Giá trị lớn nhất là a[4] = 90, hoán đổi a[1] vs a[4], ta được:

40 90 20 60 70 10 50 30

Giảm i xuống i=0, so sánh a[0] vs a[1], a[2]. Giá trị lớn nhất là a[1]=90, hoán đổi a[0] và a[1], ta được:

90 40 20 60 70 10 50 30

Xét lại tính lan truyền tại vị trí a[1]. Giá trị lớn nhất của a[1], a[3],a[4] là a[3] = 60, hoán đổi a[3] vs a[1], ta được:

90 60 20 40 70 10 50 30

Thực hiện hoán đổi a[0] và a[n-1], ta được kết quả cuối cùng như sau:

30 60 20 40 70 10 50 90

Tạo lại dãy từ a[0] đến a[6] và tiếp tục thực hiện như trên.

KQ: 40 30 20 50 10 90 60 70

Độ phức tạp của thuật toán trên thuộc lớp O(n log n)