BÀI 2 CHƯƠNG 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **40** | **70** | **20** | **60** | **90** | **10** | **50** | **30** |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

**2.1 INSERTION SORT**

**B1**: Giả sử phần tử đầu tiên đã có thứ tự

40 70 20 60 90 10 50 30

i=1 Xét phần tử a[1] = 70 => a[1] = 70 > a[0] = 40

=>Chèn a[1] bên phải a[0]

40 70 20 60 90 10 50 30

**B2**: i=2 Xét phần tử a[2] = 20 => a[2] = 20 < a[0] = 40

=>Chèn a[2] bên trái a[0]

20 40 70 60 90 10 50 30

**B3**: i=3 Xét phần tử a[3] = 60 => a[3] = 60 < a[2] = 70

=>Chèn a[3] bên trái a[2]

20 40 60 70 90 10 50 30

**B4**: i=4 Xét phần tử a[4] = 90 => a[4] = 90 > a[3] = 70

=>Chèn a[4] bên phải a[3]

20 40 60 70 90 10 50 30

**B5**: i=5 Xét phần tử a[5] = 10 => a[5] = 10 < a[0] = 20

=>Chèn a[5] bên trái a[0]

10 20 40 60 70 90 50 30

**B6**: i=6 Xét phần tử a[6] = 50 => a[6] = 50 > a[2] =40

=>Chèn a[6] bên phải a[2]

10 20 40 50 60 70 90 30

**B7**: i=7 Xét phần tử a[7] = 30 => a[7] = 30 > a[1] = 20

=>Chèn a[7] bên phải a[1]

10 20 30 40 50 60 70 90

ĐÁP ÁN: 10 20 30 40 50 60 70 90

Độ phức tạp của thuật toán: O(n2)

**2.2 SELECTION SORT**

**B1**: i=0, duyệt 0 -> 7

40 70 20 60 90 10 50 30

->min\_pos=5

->swap a[0] với a[5]

-> 10 70 20 60 90 40 50 30

**B2**: i=1, duyệt 1 -> 7

10 70 20 60 90 40 50 30

->min\_pos=2

->swap a[1] với a[2]

-> 10 20 70 60 90 40 50 30

**B3**: i=2, duyệt 2 -> 7

10 20 70 60 90 40 50 30

->min\_pos=7

->swap a[2] với a[7]

-> 10 20 30 60 90 40 50 70

**B4**: i=3, duyệt 3 -> 7

10 20 30 60 90 40 50 70

->min\_pos=5

->swap a[3] với a[5]

-> 10 20 30 40 90 60 50 70

**B5**: i=4, duyệt 4 -> 7

10 20 30 40 90 60 50 70

->min\_pos=6

->swap a[4] với a[6]

-> 10 20 30 40 50 60 90 70

**B6**: i=5, duyệt 5 -> 7

10 20 30 40 50 60 90 70

->min\_pos=5

-> không swap

-> 10 20 30 40 50 60 90 70

**B7**: i=6, duyệt 6 -> 7

10 20 30 40 50 60 90 70

->min\_pos=6

-> swap a[6] với a[7]

-> 10 20 30 40 50 60 70 90

Độ phức tạp của thuật toán: O(n2)

**2.3 INTERCHANGE SORT**

\*i=0

j=i+1= 1, a[0] <a[1] >>>>ko swap

j=2, a[0]>a[2] >>>>>swap

1. 70 40 60 90 10 50 30

j=3, j=4, a[0] < a[3] và a[4] >>>>ko swap

j=5, a[0] > a[5] >>>>>swap

1. 70 40 60 90 20 50 30

j=6, j=7, a[0] < a[6] và a[7] >>>>ko swap

10 70 40 60 90 20 50 30

Tăng i lên 1 đơn vị

\*i=1

j=2, a[1] > a[2] >>>>swap

10 40 70 60 90 20 50 30

j=3, a[1] <a[3] >>>>ko swap

j=4 >>>>swap

10 20 70 60 90 40 50 30

j=5,6,7, a[1] < a[5], a[6],a[7] >>>>ko swap

10 20 70 60 90 40 50 30

\*i=2

j=3, swap

10 20 60 70 90 40 50 30

j=4, ko swap

j=5, swap

10 20 40 70 90 60 50 30

j=6, ko swap

j=7, swap

10 20 30 70 90 60 50 40

Thực hiện tăng i cho đến khi dãy số về thứ tự tăng dần

10 20 30 40 50 60 70 90

Với thứ tự giảm dần, thực hiện ngược lại.

Độ phức tạp của thuật toán: O(n2)

**2.4 BUBBLE SORT**

* **j = 7**, vì a[j] ( a[7] ) bé hơn a[j-1] ( a[6] )nên đổi chỗ giá trị giữa a[7] và a[6].

Phần tử: 40 70 20 60 90 10 **30 50**

Vị trí: 0 1 2 3 4 5 6 7

* **j = 6**, vì a[6] lớn hơn a[5] nên không đổi chỗ.

Phần tử: 40 70 20 60 90 **10 30** 50

Vị trí: 0 1 2 3 4 5 6 7

* **j = 5**, vì a[5] bé hơn a[4] nên đổi chỗ giá trị giữa a[5] và a[4].

Phần tử: 40 70 20 60 **10 90** 30 50

Vị trí: 0 1 2 3 4 5 6 7

* **j = 4**, vì a[4] bé hơn a[3] nên đổi chỗ giá trị giữa a[4] và a[3].

Phần tử: 40 70 20 **10** **60** 9030 50

Vị trí: 0 1 2 3 4 5 6 7

* **j = 3**, vì a[3] lớn hơn a[2] nên đổi chỗ giá trị giữa a[3] và a[2].

Phần tử: 40 70 **10 20** 60 9030 50

Vị trí: 0 1 2 3 4 5 6 7

* **j = 2**, vì a[2] bé hơn a[1] nên đổi chỗ giá trị giữa a[2] và a[1].

Phần tử: 40 **10 70** 20 60 9030 50

Vị trí: 0 1 2 3 4 5 6 7

* **j = 1**, vì a[1] bé hơn a[0] nên đổi chỗ giá trị giữa a[1] và a[0].

Phần tử: **10 40** 70 20 60 9030 50

Vị trí: 0 1 2 3 4 5 6 7

* **j = 7**, vì a[7] lớn hơn a[6] nên không đổi chỗ.

Phần tử: 10 40 70 20 60 90 **30 50**

Vị trí: 0 1 2 3 4 5 6 7

* **j = 6**, vì a[6] bé hơn a[5] nên đổi chỗ giá trị giữa a[6] và a[5].

Phần tử: 10 40 70 20 60 **30 90** 50

Vị trí: 0 1 2 3 4 5 6 7

* **j = 5**, vì a[5] bé hơn a[4] nên đổi chỗ giá trị giữa a[5] và a[4].

Phần tử: 10 40 70 20 **30** **60** 90 50

Vị trí: 0 1 2 3 4 5 6 7

* **j = 4**, vì a[4] lớn hơn a[3] nên không đổi chỗ.

Phần tử: 10 40 70 **20 30** 6090 50

Vị trí: 0 1 2 3 4 5 6 7

* **j = 3**, vì a[3] bé hơn a[2] nên đổi chỗ giá trị giữa a[3] và a[2].

Phần tử: 10 40 **20 70** 30 6090 50

Vị trí: 0 1 2 3 4 5 6 7

* **j = 2**, vì a[2] bé hơn a[1] nên đổi chỗ giá trị giữa a[2] và a[1].

Phần tử: 10 **20** **40** 70 30 6090 50

Vị trí: 0 1 2 3 4 5 6 7

* **j = 1**, vì a[1] lớn hơn a[0] nên không đổi chỗ.

Phần tử: **10 20** 40 70 30 6090 50

Vị trí: 0 1 2 3 4 5 6 7

* **j = 7**, vì a[7] bé hơn a[6] nên đổi chỗ giá trị giữa a[7] và a[6].

Phần tử: 10 20 40 70 30 60 **50 90**

Vị trí: 0 1 2 3 4 5 6 7

* **j = 6**, vì a[6] bé hơn a[5] nên đổi chỗ giá trị giữa a[6] và a[5].

Phần tử: 10 20 40 70 30 **50 60** 90

Vị trí: 0 1 2 3 4 5 6 7

* **j = 5**, vì a[5] lớn hơn a[4] nên không đổi chỗ.

Phần tử: 10 20 40 70 **30 50** 6090

Vị trí: 0 1 2 3 4 5 6 7

* **j = 4**, vì a[4] bé hơn a[3] nên đổi chỗ giá trị giữa a[4] và a[3].

Phần tử: 10 20 40 **30** **70** 50 6090

Vị trí: 0 1 2 3 4 5 6 7

* **j = 3**, vì a[3] bé hơn a[2] nên đổi chỗ giá trị giữa a[3] và a[2].

Phần tử: 10 20 **30** **40** 7050 6090

Vị trí: 0 1 2 3 4 5 6 7

* **j = 2**, vì a[2] lớn hơn a[1] nên không đổi chỗ.

Phần tử: 10 **20** **30** 40 7050 6090

Vị trí: 0 1 2 3 4 5 6 7

* **j = 7**, vì a[7] lớn hơn a[6] nên không đổi chỗ.

Phần tử: 10 20 30 40 7050 **60 90**

Vị trí: 0 1 2 3 4 5 6 7

* **j = 6**, vì a[6] lớn hơn a[5] nên không đổi chỗ.

Phần tử: 10 20 30 40 70 **50 60** 90

Vị trí: 0 1 2 3 4 5 6 7

* **j = 5**, vì a[5] bé hơn a[4] nên đổi chỗ giá trị giữa a[5] và a[4].

Phần tử: 10 20 30 40 **50 70** 6090

Vị trí: 0 1 2 3 4 5 6 7

* **j = 4**, vì a[4] bé hơn a[3] nên đổi chỗ giá trị giữa a[4] và a[3].

Phần tử: 10 20 30 **40** **50** 706090

Vị trí: 0 1 2 3 4 5 6 7

* **j = 7**, vì a[7] lớn hơn a[6] nên không đổi chỗ.

Phần tử: 10 20 30 40 5070 **60 90**

Vị trí: 0 1 2 3 4 5 6 7

* **j = 6**, vì a[6] bé hơn a[5] nên đổi chỗ giá trị giữa a[6] và a[5].

Phần tử: 10 20 30 40 50 **60 70** 90

Vị trí: 0 1 2 3 4 5 6 7

* **j = 5**, vì a[5] lớn hơn a[4] nên không đổi chỗ.

Phần tử: 10 20 30 40 **50 60** 7090

Vị trí: 0 1 2 3 4 5 6 7

- Độ phức tạp của thuật toán: O()

**2.5 QUICK SORT**

L=0, r=7

Chọn x= a[3] = 60

Bước 1:

i=0 , a[0] = 40 < x=60 -> i++

i=1 , a[1] = 70 > x=60 -> dừng

j=7 , a[7] = 30 < x=60 -> dừng

->Swap a[i] với a[j] ( a[1] với a[7] )

🡺 40, 30, 20, 60, 90, 10, 50, 70

* i++ , j- - ( i=2 , j=6)

Bước 2:

i=2 , a[2] = 20 < x=60 -> i++

i=3 , a[3] = 60 < x=60 -> dừng

j=6 , a[6] = 50 < x=60 -> dừng

->Swap a[i] với a[j] ( a[3] với a[6] )

🡪 40, 30, 20, 50, 90, 10, 60, 70

* i++ , j- - ( i=4 , j=5)

Bước 3:

i=4 , a[4]=90 > x= 60 -> dừng

j=5 , a[5]=10 < x=60 -> dừng

->Swap a[i] với a[j] ( a[4] với a[5] )

🡪 40, 30, 20, 50, 10, 90, 60, 70

* i++ , j- - ( i=5 , j= 4) 🡪 dừng ( i>j )

\*Ta có 2 dãy con: 40, 30, 20, 50, 10 ( dãy 1)

90, 60, 70 (dãy 2)

**Dãy 1: 40, 30, 20, 50, 10**

L=0 , r=4

Chọn x = a[2] = 20

Bước 1:

i = 0 , a[0] = 40 > x=20 🡪 dừng

j = 4 , a[4] = 10 < x = 20 🡪 dừng

* Swap a[i] với a[j] ( a[0] với a[4] )
* 10, 30, 20, 50, 40
* i ++ , j- - ( i=1 , j= 3)

Bước 2:

i = 1 , a[1] = 30 > x=20 -> dừng

j = 3 , a[3] = 50 > x=20 -> j- -

j = 2 , a[2] = 20 > x=20 -> dừng

* Swap a[i] với a[j] ( a[1] với a[2] )
* 10, 20, 30, 50, 40
* i++, j- - ( i = 2 , j = 1) 🡺 dừng ( i>j )

10, 20, 30, 50, 40

Ta có 2 dãy con: 10, 20 ( dãy 1\*)

30, 50, 40 (dãy 2\*)

Dãy 1\*: 10, 20

L=0 , r=1

Chọn x=a[0]=10

Bước 1:

i = 0 , a[0] = 10 < x = 10 -> dừng

j = 1 , a[1] = 20 > x = 10 -> j++

i ++ = 1 , j- - = 0 ( i> j) 🡺 dừng ( không swap )

* 10, 20 (\*)

Dãy 2\*: 30, 50, 40

L=0 , r= 2

Chọn x= a[1] = 50

Bước 1:

i=0, a[0] = 30 < x=50 -> i++;

i=1, a[1] = 50 < x=50 -> dừng

j=2 , a[2] = 40 < x=50 -> dừng

🡺Swap a[i] với a[j] ( a[1] với a[2] )

🡺30, 40 , 50

* i++ = 2 , j- - = 1 🡺 dừng ( i>j )
* 30, 40, 50 (\*\*)

Từ (\*) và (\*\*) 🡺 10, 20, 30, 40, 50 (+)

Dãy 2: 90, 60, 70

L=0 , r = 2

Chọn x = a[1] = 60

Bước 1:

i=0 , a[0] = 90 > x=60 -> dừng

j=2 , a[2] = 70 > x=60 -> j- -

j=1 , a[1] = 60 > x=60 -> dừng

🡺Swap a[i] với a[j] ( a[0] với a[1] )

* 60, 90, 70
* i++ = 1 , j- - = 0 🡺 dừng ( i>j )
* 60, 90, 70

Ta có 2 dãy con: 60 (1) 🡺 đã có thứ tự

90, 70 (2)

(2) 🡺 90, 70

L= 0 , r= 1

Chọn x = a[0] = 90

Bước 1:

i=0 , a[0] = 90 < x= 90 -> dừng

j=1 , a[1] = 70 < x= 90 –> dừng

🡺Swap a[i] với a[j] ( a[0] với a[1] )

* 70, 90

Từ dãy 2 🡺 60, 70, 90 (++)

**Từ (+) và (++) 🡺 10, 20, 30, 40, 50, 60, 70, 90**

**2.6 HEAP SORT**

40 70 20 60 90 10 50 30

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

Bước 1:

Dãy 1: a[0] -> a[n\2-1]=a[3]

Dãy 2: a[n\2] -> a[n-1]

a[4] -> a[7]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **40** | **70** | **20** | **60** | 90 | 10 | 50 | 30 |

* Xét i=3 a[3]=60

a[7]=30

a[6]=50

* Không swap
* 40 70 20 60 90 10 50 30
* Xét i=2 a[2]=20

a[5]=10

a[4]=90

* Swap a[2] và a[4]->heap max
* 40 70 90 60 20 10 50 30
* Xét i=1 a[1]=70

a[3]=60

a[2]=90

* Swap a[1] và a[2]->heap max
* 40 90 70 60 20 10 50 30
* Xét i=0 a[0]=40

a[1]=90

a[2]=70

* Swap a[0] và a[1]->heap max
* 90 40 70 60 20 10 50 30
* Xét i=1 a[1]=40

a[3]=60

a[2]=70

* Swap a[1] và a[2]->heap max
* 90 70 40 60 20 10 50 30
* Xét i=2 a[2]=40

a[4]=20

a[5]=10

* Không swap(dừng i)
* 90 70 40 60 20 10 50 30
* 30 70 40 60 20 10 50 90

Bước 2:

30 70 40 60 20 10 50

a[0] a[1] a[2] a[3] a[4] a[5] a[6]

Dãy 1: a[0] -> a[n\2-1]=a[2]

Dãy 2: a[n\2] -> a[n-1]

a[3] -> a[6]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **30** | **70** | **40** | **60** | 20 | 10 | 50 |

* Xét i=3 a[3]=60

a[5]=10

a[6]=50

* Không swap
* 30 70 40 60 20 10 50
* Xét i=2 a[2]=40

a[4]=20

a[3]=60

* Swap a[2] và a[3]->heap max
* 30 70 60 40 20 10 50
* Xét i=1 a[1]=70

a[2]=60

a[3]=40

* Không swap
* 30 70 60 40 20 10 50
* Xét i=0 a[0]=30

a[1]=70

a[2]=60

* Swap a[0] và a[1]->heap max
* 70 30 60 40 20 10 50
* Xét i=1 a[1]=30

a[2]=60

a[3]=40

* Swap a[1] và a[2]->heap max
* 70 60 30 40 20 10 50
* Xét i=2 a[2]=30

a[4]=20

a[3]=40

* Swap a[3] và a[2]->heap max
* 70 60 40 30 20 10 50
* Xét i=3 a[3]=30

a[5]=10

a[6]=50

* Swap a[6] và a[3]->heap max
* 70 60 40 50 20 10 30
* Xét i=2 a[2]=40

a[4]=20

a[3]=50

* Swap a[2] và a[3]->heap max
* 70 60 50 40 20 10 30
* Xét i=1 a[1]=60

a[2]=50

a[3]=40

* Không swap(dừng i)
* 70 60 40 50 20 10 30
* 30 60 40 50 20 10 70

Bước 3:

30 60 40 50 20 10

a[0] a[1] a[2] a[3] a[4] a[5]

Dãy 1: a[0] -> a[n\2-1]=a[2]

Dãy 2: a[n\2] -> a[n-1]

a[3] -> a[5]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **30** | **60** | **40** | 50 | 20 | 10 |

* Xét i=2 a[2]=40

a[4]=20

a[5]=10

* Không swap
* 30 60 40 50 20 10
* Xét i=1 a[1]=60

a[2]=40

a[3]=50

* Không swap
* 30 60 40 50 20 10
* Xét i=0 a[0]=30

a[1]=60

a[2]=40

* Swap a[0] và a[2]->heap max
* 60 30 40 50 20 10
* Xét i=1 a[1]=30

a[2]=40

a[3]=50

* Swap a[1] và a[3]->heap max
* 60 50 40 30 20 10
* Xét i=2 a[2]=40

a[4]=20

a[5]=10

* Không swap(dừng i)
* 60 50 40 30 20 10

10 50 40 30 20 60

Bước 4:

10 50 40 30 20

a[0] a[1] a[2] a[3] a[4]

Dãy 1: a[0] -> a[n\2-1]=a[1]

Dãy 2: a[n\2] -> a[n-1]

a[2] -> a[4]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **10** | **50** | 40 | 30 | 20 |

* Xét i=1 a[1]=10

a[3]=30

a[4]=20

* Swap a[1] và a[3]->heap max
* 30 50 40 10 20
* Xét i=0 a[0]=30

a[1]=50

a[2]=40

* Swap a[0] và a[1]->heap max
* 50 30 40 10 20
* Xét i=1 a[1]=30

a[3]=10

a[4]=20

* Không swap(dừng i)
* 50 30 40 10 20
* 20 30 40 10 50

Bước 4:

20 30 40 10

a[0] a[1] a[2] a[3]

Dãy 1: a[0] -> a[n\2-1]=a[1]

Dãy 2: a[n\2] -> a[n-1]

a[2] -> a[3]

|  |  |  |  |
| --- | --- | --- | --- |
| **20** | **30** | 40 | 10 |

* Xét i=1 a[1]=30

a[2]=40

a[3]=10

* Swap a[1] và a[2]->heap max
* 20 40 30 10
* Xét i=0 a[0]=20

a[2]=30

a[1]=40

* Swap a[1] và a[0]->heap max
* 40 20 30 10
* Xét i=1 a[1]=20

a[2]=30

a[3]=10

* Swap a[1] và a[2]->heap max
* 40 30 20 10
* Xét i=0 a[0]=40

a[2]=20

a[1]=30

* Không swap(dừng i)
* 40 30 20 10
* 10 30 20 40

Bước 5:

10 30 20

a[0] a[1] a[2]

Dãy 1: a[0] -> a[n\2-1]=a[0]

Dãy 2: a[n\2] -> a[n-1]

a[1] -> a[2]

|  |  |  |
| --- | --- | --- |
| **10** | 30 | 20 |

* Xét i=0 a[0]=10

a[2]=20

a[1]=30

* Swap a[1] và a[0]->heap max
* 30 10 20
* 30 10 20
* 20 10 30

Bước 6:

20 10

a[0] a[1]

|  |  |
| --- | --- |
| **20** | 10 |

Dãy 1: a[0] -> a[n\2-1]=a[0]

Dãy 2: a[n\2] -> a[n-1]

a[1] -> a[1]

* Xét i=0 a[0]=20

a[1]=10

* Swap a[1] và a[0]->heap max
* 10 20
* 10 20
* Danh sách sắp xếp tăng dần theo phương pháp Heap Sort là:
* 10 20 30 40 50 60 70 90