**Ôn tập phần SẮP XẾP & TÌM KIẾM**

**Câu 1:** Hãy trình bày các bước sắp xếp dãy số {7, 2, 5, 9, 1} theo thứ tự tăng dần dùng thuật toán Bubble Sort.

B1: i=0

j=4, xét a[4] = 1 < a[3] = 9 => swap

7, 2, 5, 1, 9

j=3, xét a[3] = 1 < a[2] = 5 => swap

7, 2, 1, 5, 9

j=2, xét a[2] = 1 < a[1] = 2 => swap

7, 1, 2, 5, 9

J=1, xét a[1] =1 < a[0] = 7 => swap

1, 7, 2, 5, 9

B2: i=1

j=4, xét a[4] = 9 > a[3] = 5 => ko swap

1, 7, 2, 5, 9

j=3, xét a[3] = 5 > a[2] = 2 => ko swap

1, 7, 2, 5, 9

j=2, xét a[2] = 2 < a[1] = 7 => swap

1, 2, 7, 5, 9

B3: i=2

j=4, xét a[4] = 9 > a[3] = 5 => ko swap

1, 2, 7, 5, 9

j=3, xét a[3] = 5 > a[2] = 7 => swap

1, 2, 5, 7, 9

B4: i=3

j=4, xét a[4] = 9 > a[3] = 5 => ko swap

1, 2, 5, 7, 9

Kq: 1, 2, 5, 7, 9

**Câu 2:** Cho dãy số nguyên như sau: 22 12 9 7 31 19

1. Trình bày các bước thuật toán Chèn trực tiếp (Insertion Sort) để sắp xếp dãy số trên tăng dần.

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

22 12 9 7 31 19

i=1 Xét phần tử a[1] = 12 => a[0] = 22 > a[1] = 12

=> Chèn a[1] vào bên trái a[0]

12 22 9 7 31 19

i=2 Xét phần tử a[2] = 9 => a[2] = 9 < a[0] = 12

=> Chèn a[2] vào bên trái a[0]

9 12 22 7 31 19

i=3 Xét phần tử a[3] = 7 => a[3] = 7 < a[0] = 9

=> Chèn a[3] vào bên trái a[0]

7 9 12 22 31 19

i=4 Xét phần tử a[4] = 31 => a[4] = 31 > a[3] = 22

=> Chèn a[1] vào bên trái a[0]

7 9 12 22 31 19

i=5 Xét phần tử a[5] = 31 => a[5] = 31 > a[4] = 19

=> Chèn a[5] vào bên phải a[4]

7 9 12 22 19 31

KQ: 7 9 12 22 19 31

(b) Trình bày ngắn gọn ý tưởng thuật toán Heap Sort.

Chia thành 2 dãy:

Từ a[0] -> a[2]

Từ a[3] -> a[5]

Xét i = 1, a[1] = 12

a[3] = 7

a[4] = 31

=> swap a[1] với a[4]

22 31 9 7 12 19

Xét i = 2, a[2] = 9

a[5] = 19

=> swap a[2] với a[5]

22 31 19 7 12 9

Xét i=0, a[0] = 22

a[1] = 31

a[2] = 19

=> swap a[0] với a[1]

31 22 19 7 12 9

9 22 19 7 12 31

Tiếp tục như thao tác trên, ta đc:

7 9 12 19 22 31

**Câu 3:** Xét mảng các số nguyên có nội dung như sau :

-9 -9 -5 -2 0 3 7 7 10 15

(a) Tính số lần so sánh để tìm ra phần tử X = -9 bằng phương pháp:

a.1 Tìm tuyến tính

Có 1 lần so sánh để tìm ra

a.2 Tìm nhị phân

Có 3 lần so sánh để tìm ra

Nhận xét và so sánh hai phương pháp tìm nêu trên trong trường hợp này và trong trường hợp tổng quát.

Trường hợp 1 có tốc độ tìm kiếm nhanh hơn. So với trường hợp 2 cần phải sắp xếp theo thứ tự sau đó mới có thể thực hiện tìm kiếm nhị phân.

(b) Trong trường hợp tìm nhị phân, phần tử nào sẽ được tìm thấy (thứ 1 hay 2)

Phần tưr thứ 2.

**Câu 4:** Xây dựng thuật toán tìm phần tử nhỏ nhất (lớn nhất) trong một mảng các số nguyên.

Lớn nhất:

*Bước 1: Nhập N và dãy a1,a2, ..., aN.*

*Bước 2: Max <-- a1, i <-- 2;*

*Bước 3: Nếu i > N thì đưa ra giá trị Max rồi kết thúc;*

*Bước 4: Nếu****ai > Max****thì Max <-- ai;*

*Bước 5: i <-- i + 1 rồi quay lại Bước 3;*

Nhỏ nhất:

*Bước 1: Nhập N và dãy a1,a2, ..., aN.*

*Bước 2: Min <-- a1, i <-- 2;*

*Bước 3: Nếu i > N thì đưa ra giá trị Min rồi kết thúc;*

*Bước 4: Nếu****ai < Min****thì Min <-- ai;*

*Bước 5: i <-- i + 1 rồi quay lại Bước 3;*

**Câu 5:** Cho dãy số nguyên như sau:

11 2 15 20 9 7 8 10

(a) Trình bày các bước thuật toán Đổi chỗ trực tiếp (Interchange Sort) để sắp xếp dãy số trên giảm dần.

B1: i = 0

j=1, a[0] = 11 > a[1] = 2 => ko swap

j=2, a[0] = 11 < a[2] = 15 => swap

15 2 11 20 9 7 8 10

j=3, a[0] = 15 < a[3] = 20 => swap

1. 2 11 15 9 7 8 10

j=4, a[0] = 20 > a[4] = 9 => ko swap

j=5, a[0] = 20 > a[5] = 7 => ko swap

j=6, a[0] = 20 > a[6] = 8 => ko swap

j=7, a[0] = 20 > a[7] = 10 => ko swap

20 2 11 15 9 7 8 10

Tương tự các bước sau như trên:

20 15 11 10 9 8 7 2

(b) Trình bày các bước thuật toán Nổi bọt (Bubble Sort) để sắp xếp dãy số trên tăng dần.

B1: i=0

j=7, xét a[7] = 10 > a[6] = 8 => ko swap

j=6, xét a[6] = 8 > a[5] = 7 => ko swap

j=5, xét a[5] = 7 < a[4] = 9 => swap

11 2 15 20 7 9 8 10

j=4, xét a[4] =7 < a[3] = 20 => swap

11 2 15 7 20 9 8 10

j=3, xét a[3] = 7 < a[2] = 15 => swap

11 2 7 15 20 9 8 10

j=2, xét a[2] = 7 < a[1] = 2 => ko swap

j=1, xét a[1] = 2 < a[0] = 11 => swap

1. 11 7 15 20 9 8 10

Tương tự các bước sau như trên:

2 7 8 9 10 11 15 20

(c) Trình bày các bước thuật toán Lựa chọn trực tiếp (Selection Sort) để sắp xếp dãy số trên giảm dần.

B1: a[0] -> a[7]

a[1] = max

Swap a[0] với a[1]

20 2 15 2 9 7 8 10

B2: a[1] -> a[7]

a[2] = max

Swap a[2] với a[1]

20 15 11 2 9 7 8 10

B3: a[2] -> a[7]

a[2] = max

Ko swap

20 15 11 2 9 7 8 10

B4: a[3] -> a[7]

a[7] = max

Swap a[7] với a[3]

20 15 11 10 9 7 8 2

B5: a[4] -> a[7]

a[4] = max

Ko swap

20 15 11 10 9 7 8 2

B6: a[5] -> a[7]

a[5] = max

Ko swap

20 15 11 10 9 *7* 8 2

B7: a[6] -> a[7]

a[6] = max

Ko swap

20 15 11 10 9 7 8 2

KQ:

20 15 11 10 9 7 8 2