**BÀI TẬP CHƯƠNG 3**

**Bài 2: Câu 2.6**

0 1 2 3 4 5 6 7

40 70 20 60 90 10 50 30

Heapsort

Chia dãy trên thành 2 đoạn, bao gồm:

+ Nửa dãy bên trái chứa các phần tử a[0] đến a[(n/2)-1]: 40 70 20 60

+ Nửa dãy bên phải chứa các phần tử a[a/2] đến a[n-1]: 90 10 50 30

0 1 2 3 4 5 6 7

40 70 20 60 | 90 10 50 30

Tại vị trí cuối cùng của dãy con bên trái i=3, so sánh a[i] = a[3] = 60 với 2 phần tử tại vị trí 2\*i+1=7 và 2\*i+2=8 (trong trường hợp này a[8]=0)

0 1 2 3 4 5 6 7

40 70 20 60 | 90 10 50 30

Ta thấy a[3] > a[7]. Không đổi

Giảm i xuống 1 giá trị (i=2), so sánh a[2] với a[5] và a[6]

1. 1 2 3 4 5 6 7
2. 70 20 60 | 90 10 50 30

Ta thấy giá trị lớn nhất của a[2], a[5] và a[6] là a[6]=50. Thự hiện hoán vị a[2] và a[6]

1. 1 2 3 4 5 6 7

40 70 50 60 | 90 10 20 30

Giảm i xuống 1 giá trị (i=1), so sánh a[1] với a[3] và a[4]

0 1 2 3 4 5 6 7

40 70 50 60 | 90 10 20 30

Ta thấy giá trị lớn nhất của a[1], a[3] và a[4] là a[4]=90. Thự hiện hoán vị a[1] và a[4]

0 1 2 3 4 5 6 7

40 90 50 60 | 70 10 20 30

Giảm i xuống 1 giá trị (i=0), so sánh a[0] với a[1] và a[2]

0 1 2 3 4 5 6 7

40 90 50 60 | 70 10 20 30

Ta thấy giá trị lớn nhất của a[0], a[1] và a[2] là a[1]=90. Thự hiện hoán vị a[0] và a[1]

0 1 2 3 4 5 6 7

90 40 50 60 | 70 10 20 30

Xét lại tính lan truyền tại vị trí a[1], sau khi a[1] nhận giá trị là 40 (thay thế giá trị 90 trước đó; giá trị a[1]=90 trước đó thỏa tính của Heap so với a[3] và a[4].

0 1 2 3 4 5 6 7

90 40 50 60 | 70 10 20 30

Ta thấy giá trị lớn nhất của a[1], a[3] và a[4] là a[4]=70. Thự hiện hoán vị a[1] và a[4]

Việc tạo Heap (Heap Max) ban đầu hoàn tất. Ta được 1 Heap sau:

0 1 2 3 4 5 6 7

90 70 50 60 40 10 20 30

Hoán vị phần tử a[0] và phẩn tử cuối Heap đang xét. Ta có kết quả sau:

0 1 2 3 4 5 6 7

30 70 50 60 40 10 20 90

Trong dãy đang xét, giới hạn phần tử cuối dãy. Ta được:

0 1 2 3 4 5 6 7

30 70 50 60 40 10 20 | 90

Tạo Heap ban đầu lại cho dãy các phần tử đang xét từ a[0] đến a[6]

Trong trường hợp này thực chất chỉ xét lại vị trí a[0] (và sự lan truyền nếu có), các vị trí còn lại từ a[1] đến a[6] đã thỏa tính chất Heap trước đó.

Sau khi từ a[0] đến a[6] là 1 Heap, hoán đổi a[0] và a[6]. Tiếp tụ cxest lại từ a[0] đến a[5]….. lặp lại cho đến khi danh sách được sắp xếp thứ tự tăng dần.

**\* Độ phức tạp: O(nlogn) = 8log8**