

# Sắp thứ tự

GV: Lê Mậu Long

## I. Các thuật giải cơ bản

Độ phức tạp  $O(n^2)$

### 1. Phương pháp chèn (Insertion sort)

Nội dung phương pháp:

- Xét dãy  $a_1, \dots, a_{i-1}$  có thứ tự, tìm vị trí thích hợp của  $a_i$  trong dãy trên để chèn vào sao cho ta được dãy  $a_1, \dots, a_i$  có thứ tự.
- Thực hiện với  $i = 2, \dots, n$  ta được dãy  $a_1, \dots, a_n$  có thứ tự

28/10/2020

GV: Lê Mậu Long

3

### Ví dụ

1	2	3	4	5	6	7	8
7	1	6	9	3	5	4	8
1	7	6					
1	6	7	9				
1	6	7	9	3			
1	3	6	7	9	5		
1	3	5	6	7	9		
		X=	5				

28/10/2020

GV: Lê Mậu Long

4

### Thuật giải chèn

- Với  $i=2, \dots, n$ , thực hiện:
  - $x = a_i$
  - $j = i-1$
  - Khi  $(j>0) \wedge (x<a_j)$ 
    - $a_{j+1}=a_j$
    - $j = j-1$
  - $a_{j+1}=x$

28/10/2020

GV: Lê Mậu Long

5

### Độ phức tạp

- Số phép gán:

$$M_{\min} = 2(n-1)$$

$$M_{\max} = 2(n-1) + \sum_{i=2}^n (i-1) = \frac{(n-1)(n+4)}{2}$$

- Số phép so sánh:

$$C_{\min} = n-1$$

$$C_{\max} = \frac{n(n-1)}{2}$$

28/10/2020

GV: Lê Mậu Long

6

### Thuật giải chèn (Java)

```
for (i=1; i<n; i++)
{
    x = a[i];
    j = i-1;
    while (j>0 && x<a[j])
    {
        a[j+1]=a[j];
        j--;
    }
    a[j+1]=x;
}
```

28/10/2020

GV: Lê Mậu Long

7

### Thực hành

- Xây dựng class thể hiện danh sách

```
class ArrList
{
    private int[] a; //mảng a chứa dữ liệu
    private int n; // số phần tử dữ liệu
    // constructor cấp phát mảng
    public ArrList (int size)
    {
        a = new int[size];
        n = 0;
    }
}
```

28/10/2020

GV: Lê Mậu Long

8

## Thực hành

```
//Thêm phần tử x vào cuối danh sách
public void add(int x)
{
    a[n++] = x;
}
//Gán x vào phần tử thứ i
public void set(int x, int i)
{
    a[i] = x;
}
```

28/10/2020

GV: Lê Mậu Long

9

## Thực hành

```
//Lấy giá trị phần tử thứ i
public int get(int i)
{
    return a[i];
}
//Trả về số phần tử
public int nItem()
{
    return n;
}
```

28/10/2020

GV: Lê Mậu Long

10

## Thực hành

```
public void insertionsort()
{
    int i, j;
    int x;
    for (i=1; i<n; i++)
    {
        x = a[i];
        j = i-1;
        while (j>=0 && x<a[j])
        {
            a[j+1] = a[j];
            j--;
        }
        a[j+1] = x;
    }
}
// Kết thúc class ArrList
}
```

28/10/2020

GV: Lê Mậu Long

11

## Thực hành

```
//Chương trình sử dụng class ArrList
public class AppArray
{
    static Scanner x;
    //phương thức nhập dữ liệu
    static void inputA(ArrList A)
    {
        int xa;
        do {
            xa= x.nextInt();
            if (xa>0)
                A.add(xa);
        } while (xa>0);
    }
}
```

28/10/2020

GV: Lê Mậu Long

12

## Thực hành

```
//phương thức xuất danh sách
static void displayA(ArrayList A)
{
    int count = 0;
    for (int i=0; i<A.nItem(); i++)
    {
        System.out.printf(" %8d", A.get(i));
        count++;
        if (count%10==0)
            System.out.println();
    }
}
```

28/10/2020

GV: Lê Mậu Long

13

## Thực hành

```
public static void main(String[] args)
{
    ArrayList A = new ArrayList(100);
    int chon;
    x = new Scanner(System.in);
    do {
        System.out.print("\n1. Nhập số liệu");
        System.out.print("\n2. In số liệu");
        System.out.print("\n3. Sắp thu tu tăng dần");
        System.out.print("\nNhập số 0 để kết thúc");
        System.out.print("\nChon :");
        chon = x.nextInt();
    }
```

28/10/2020

GV: Lê Mậu Long

14

```
switch (chon)
{
    case 1:
        System.out.print("\nNhập dãy 1(0:kết thúc):");
        inputA(A);
        break;
    case 2:
        System.out.println("\nDãy số:");
        displayA(A);
        break;
    case 3:
        A.insertionsort();
        break;
}
} while (chon>0);
} //hết hàm main
} //hết class AppArray
```

28/10/2020

GV: Lê Mậu Long

15

## 2. Phương pháp chọn (Selection sort)

Nội dung phương pháp:

- Chọn phần tử nhỏ nhất trong dãy  $a_1, \dots, a_n$  là  $a_k$
- Hoán đổi phần tử  $a_i$  với  $a_k$ .

Thực hiện với  $i = 1, \dots, n-1$  ta được dãy  $a_1, \dots, a_n$  có thứ tự

28/10/2020

GV: Lê Mậu Long

16

## Ví dụ

1	2	3	4	5	6	7	8
7	1	6	9	3	5	4	8
1	7	6	9	3	5	4	8
1	3	6	9	7	5	4	8
1	3	4	9	7	5	6	8
1	3	4	5	7	9	6	8
1	3	4	5	6	9	7	8
	...						

28/10/2020

GV: Lê Mậu Long

17

## Thuật giải chọn

- Với  $i=1, \dots, n-1$ , thực hiện:
  - $k=i$
  - $x = a_k$
  - Với  $j = i+1, \dots, n$ , thực hiện:
    - Nếu  $x > a_j$ 
      - $k=j$
      - $x=a_k$
  - $a_k = a_i$
  - $a_i=x$

28/10/2020

GV: Lê Mậu Long

18

## Độ phức tạp

- Số phép gán:

$$M_{\min} = 3(n-1)$$

$$M_{\max} = 3(n-1) + \sum_{i=2}^n (i-1) = \frac{(n-1)(n+6)}{2}$$

- Số phép so sánh:

$$C_{\min} = n$$

$$C_{\max} = \frac{n(n+1)}{2}$$

28/10/2020

GV: Lê Mậu Long

19

## Cài đặt

```

void selectionsort()
{
    int i, j, k;
    int x;
    for (i=0; i<n-1; i++)
    {
        k = i;
        x = a[k];
        for (j=i+1; j<n; j++)
            if (x > a[j])
            {
                k = j;
                x = a[k];
            }
        a[k] = a[i];
        a[i] = x;
    }
}
    
```

28/10/2020

GV: Lê Mậu Long

20

### Cài đặt

```
void selectionsort()
{
    int i, j, k;
    int x;
    for (i=0; i<n-1; i++)
    {
        k = i;
        for (j=i+1; j<n; j++)
            if (a[k]>a[j])
                k = j;

        x = a[k];
        a[k] = a[i];
        a[i] = x;
    }
}
```

28/10/2020

GV: Lê Mậu Long

21

### 3. Phương pháp đổi chỗ

Nội dung phương pháp:

Duyệt qua dãy, nếu 2 phần tử kề nhau không thứ tự thì hoán đổi

Thực hiện cho đến khi dãy có thứ tự

Thuật giải nổi bọt (Bubble sort)

Phép duyệt: duyệt từ cuối về đầu dãy -> phần tử nhỏ nhất về đầu dãy: nổi bọt

28/10/2020

GV: Lê Mậu Long

22

### Thuật giải (Java)

```
for (i=1; i<n; i++)
    for (j=n-1; j>=i; j--)
        if (a[j]<a[j-1])
            swap(a[j], a[j-1]);
```

// Phải viết hàm swap(x, y) hoán vị x và y

28/10/2020

GV: Lê Mậu Long

23

```
void bubblesort()
{
    int i, j;
    for (i=1; i<n; i++)
        for (j=n-1; j>=i; j--)
            if (a[j]<a[j-1])
            {
                int t = a[j];
                a[j] = a[j-1];
                a[j-1] = t;
            }
}
```

28/10/2020

GV: Lê Mậu Long

24

## Thuật giải nổi bọt có sử dụng cờ

```

void bubblesort(OK)
{
    int i, j, OK;
    i = 1;
    do {
        OK = 1;
        for (j = n - 1; j >= i; j--)
            if (a[j] < a[j - 1])
            {
                int t = a[j];
                a[j] = a[j - 1];
                a[j - 1] = t;
                OK = 0;
            }
        i++;
    } while (!OK);
}

```

28/10/2020

GV: Lê Mậu Long

25

## Thuật giải Sàng (Shaker sort) – Đọc thêm

Nội dung phương pháp:

Thực hiện lặp 2 quá trình liên tiếp:

Duyệt từ phải qua trái: Nổi bọt

Duyệt từ trái qua phải: Lắng đọng

Mỗi quá trình ghi nhận vị trí hoán đổi sau cùng làm điểm xuất phát cho quá trình tiếp theo

Thực hiện cho đến khi phải vượt trái

28/10/2020

GV: Lê Mậu Long

26

## Thuật giải Shaker sort(Java)

```

l = 1;
r = k = n - 1;
do {
    for (j = r; j >= l; j--)
        if (a[j] < a[j - 1])
        {
            int t = a[j];
            a[j] = a[j - 1];
            a[j - 1] = t;
            k = j;
        }
    l = k + 1;
    for (j = l; j <= r; j++)
        if (a[j] < a[j - 1])
        {
            int t = a[j];
            a[j] = a[j - 1];
            a[j - 1] = t;
            k = j;
        }
    r = k - 1;
} while (l <= r);

```

28/10/2020

GV: Lê Mậu Long

27

## II. Các thuật giải cải tiến

Độ phức tạp  $O(n \lg(n))$

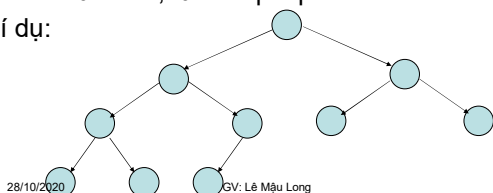
## 1. Thuật giải “vun đống” (Heap sort)

Cải tiến từ *phương pháp chọn*

Định nghĩa:

- Heap: Là cấu trúc cây nhị phân đầy đủ theo nghĩa: các nút được bố trí tuần tự từ mức thấp đến mức cao, từ trái qua phải

Ví dụ:



28/10/2020

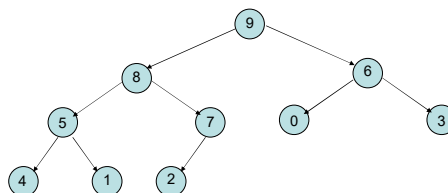
GV: Lê Mậu Long

29

## Thuật giải “vun đống” (Heap sort)

- Heap max (min): Là cấu trúc heap thỏa điều kiện : mọi nút đều có khóa lớn (nhỏ) hơn 2 con

Ví dụ:



28/10/2020

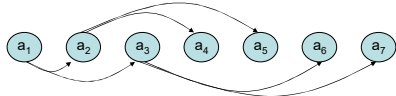
GV: Lê Mậu Long

30

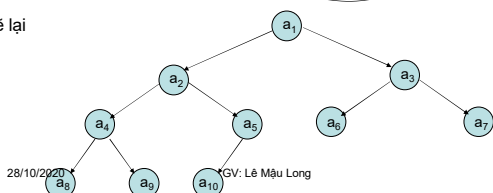
## Tổ chức heap từ dãy

- Với dãy  $a_1, a_2, \dots, a_n$  ta tổ chức thành cấu trúc heap như sau:

$a_i$  có 2 con là  $a_{2i}$  và  $a_{2i+1}$



Vẽ lại



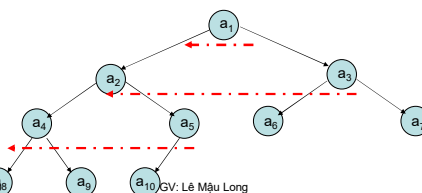
28/10/2020

GV: Lê Mậu Long

31

## Khởi tạo heap max ban đầu

- Ta có  $a_i$  với  $i=n/2+1, \dots, n$  là các nút lá trên cây nên hiển nhiên thỏa mãn tính chất heap max
- Để khởi tạo heap max ban đầu ta tìm cách **biến đổi**  $a_i$  (với  $i=n/2, \dots, 1$ ) thành heap max khi  $a_{i+1}, \dots, a_n$  thỏa heap max



28/10/2020

GV: Lê Mậu Long

32



## Sắp thứ tự

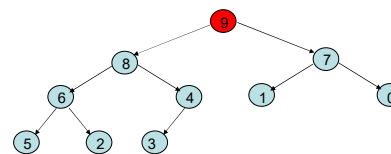
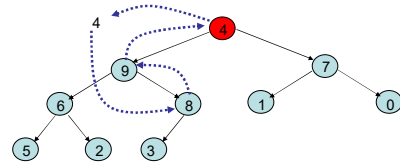
- Khi  $i=1$  ta có dãy thỏa mãn tính chất heap max, nghĩa là  $a_1$  là phần tử lớn nhất
- Để sắp thứ tự, phần tử lớn nhất phải nằm cuối nên ta hoán đổi phần tử  $a_1$  với  $a_n$
- Biến đổi  $a_1$**  thành heap max khi  $a_2, \dots, a_{n-1}$  thỏa heap max
- Tiếp tục hoán đổi và biến đổi cho đến khi dãy có thứ tự

28/10/2020

GV: Lê Mậu Long

33

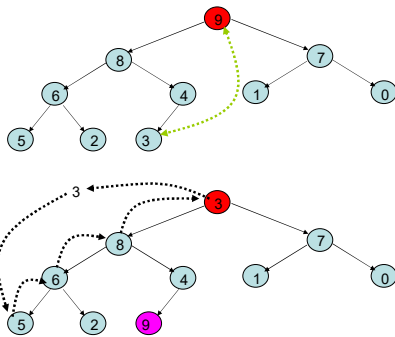
## Thuật giải biến đổi heap max (q, r)



28/10/2020

GV: Lê Mậu Long

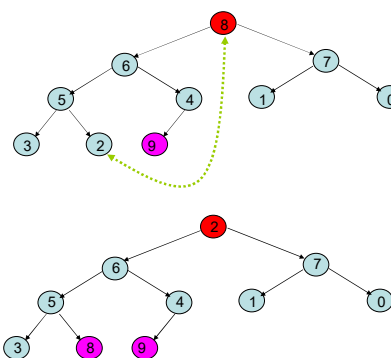
34



28/10/2020

GV: Lê Mậu Long

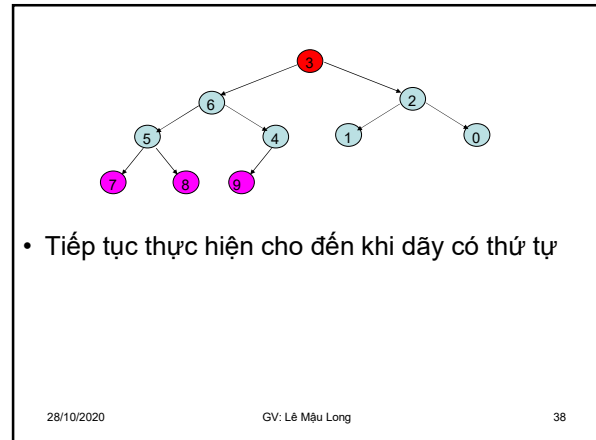
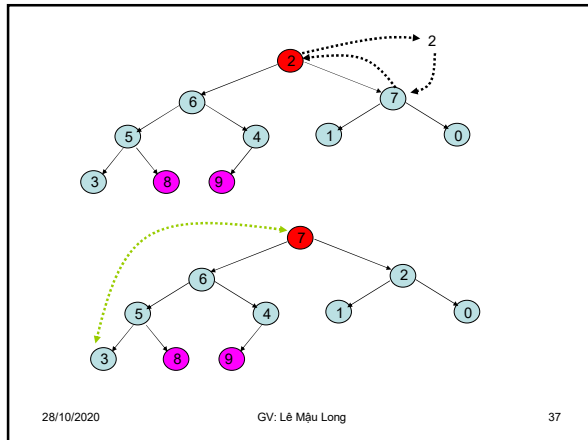
35



28/10/2020

GV: Lê Mậu Long

36



### Thuật giải biến đổi heap max (q, r)

- Khởi tạo
  - $x = a[q]$
  - $i = q$
  - $j = 2i$
  - $cont = true$
- Khi  $(j \leq r) \wedge cont$ , thực hiện:
  - Nếu  $(j < r) \wedge (a[j] < a[j+1])$  thì  $j = j+1$
  - Nếu  $x < a[j]$  thì
    - $a[i] = a[j]$
    - $i = j$
    - $j = 2i$
  - Ngược lại
    - $cont = False$
- Đổi  $a[i]$  và  $x$

28/10/2020 GV: Lê Mậu Long 39

### Sắp thứ tự

- Khởi tạo heap max ban đầu  
Với  $i = n/2, \dots, 1$ . Thực hiện:  
Biến đổi heap max(i, n)
- Sắp thứ tự
  - $i = n$
  - Lặp
    - Hoán đổi  $a_1$  với  $a_i$
    - $i = i-1$
    - Biến đổi heap max(1, i)

Cho đến khi  $i = 1$

28/10/2020 GV: Lê Mậu Long 40

```
void sift(int q, int r)
{
    int x;
    int i, j, cont = 1;
    i = q;
    j = 2*i+1;           //lưu ý
    x = a[j];
    while (j<=r && cont)
    {
        if (j<r && a[j] < a[j+1])    j++;
        if (x < a[j])
        {
            a[j] = a[j];
            i = j;
            j = 2*i+1;           //lưu ý
        }
        else
            cont = 0;
    }
    a[i] = x;
}
28/10/2020
```

GV: Lê Mậu Long

41

## Hàm Heapsort

```
void heapsort()
{
    int i;
    int x;
    if (n==0) return;
    for (i=n/2-1; i>=0; i--)           //lưu ý n/2-1 và 0
        sift(i, n-1);
    i = n-1;
    do {
        x = a[0];           //lưu ý a[0]
        a[0] = a[i];
        a[i] = x;
        i--;
        sift(0, i);           //0
    } while (i>0);
}
28/10/2020
```

GV: Lê Mậu Long

42

## 2. Thuật giải Quick sort

- Cải tiến từ phương pháp đổi chỗ
- Dựa trên phép phân hoạch:
- Chọn giá trị phân hoạch x thỏa:
 
$$\text{Min} \{ a_i \} \leq x \leq \text{Max} \{ a_i \}$$
  - Dò tìm từ trái (l) qua phải phần tử  $a_i \geq x$
  - Dò tìm từ phải (r) qua trái phần tử  $a_i \leq x$
  - Nếu  $i < j$  thì hoán đổi  $a_i$  với  $a_j$
- Tiếp tục thực hiện cho đến khi  $i > j$ 
  - Kết quả dãy sẽ bị phân hoạch thành 2 dãy:
    - Dãy từ l đến j mang giá trị  $\leq x$
    - Dãy từ i đến r mang giá trị  $\geq x$
- Để sắp thứ tự ta tiếp tục phân hoạch trên 2 đoạn (l, j) và (i, r)

GV: Lê Mậu Long

43

```
void sort(int l, int r)
{
    int i, j;
    static int x, t;
    i = l;
    j = r;
    x = a[(l+r)/2];
    do {
        while (a[i] < x) i++;
        while (a[j] > x) j--;
        if (i <= j)
        {
            t = a[i];
            a[i] = a[j];
            a[j] = t;
            i++;
            j--;
        }
    } while (i <= j);
    if (l < j) sort(l, j);
    if (i < r) sort(i, r);
}
28/10/2020
```

GV: Lê Mậu Long

44

Lời gọi

```
sort(0, n-1); // !!!
```

28/10/2020

GV: Lê Mậu Long

45

```
void qsort()
{
    sort(0, n-1);
}
```

Lời gọi

```
qsort();
```

28/10/2020

GV: Lê Mậu Long

46

## Thuật giải Shell sort – đọc thêm

Cải tiến từ phương pháp chèn trực tiếp

Nội dung:

- Chia dãy thành  $h$  dãy con bao gồm các phần tử cách nhau  $h$  vị trí
- Sắp xếp các dãy con này bằng phương pháp chèn
- Thực hiện với các bước  $h$  giảm dần về 1  
 $h_1 > h_2 > \dots > h_t = 1$

28/10/2020

GV: Lê Mậu Long

47

## Thuật giải Shell sort (Java)

```
int h[] = {15, 7, 3, 1}, t=4;
for (m=0; m<t; m++) //lặp với t dãy h
{
    k = h[m];
    for (i=k; i<n; i++)
    {
        x = a[i];
        j = i-k;
        while (j>=0 && x<a[j])
        {
            a[j+k]=a[j];
            j = j-k;
        }
        a[j+k]=x;
    }
}
```

28/10/2020

GV: Lê Mậu Long

48