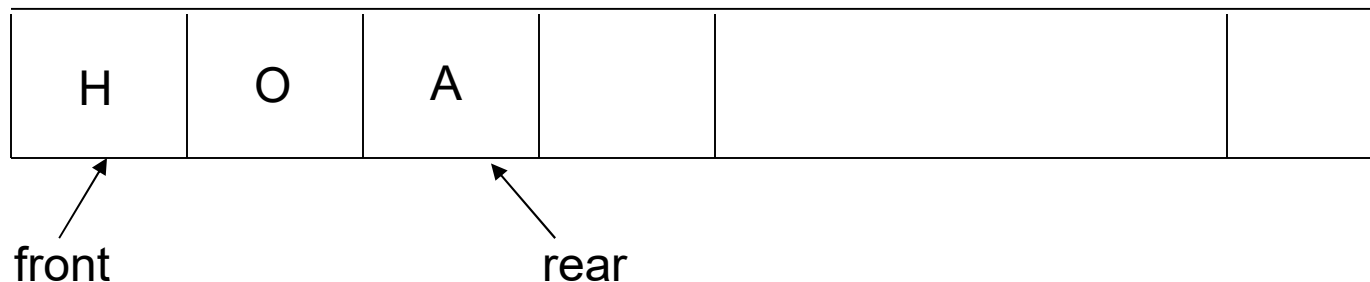


*Hàng đợi*  
*(Queue)*

# Hàng đợi (Queue)

- Là cấu trúc bao gồm các phần tử được truy xuất theo nguyên tắc “vào trước, ra trước” (First In, First Out – FIFO)

Ví dụ: Các vùng đệm giao tiếp giữa máy tính và các thiết bị



# Các thao tác trên Queue

## 4 thao tác cơ bản trên queue

- Khởi tạo queue rỗng: *QueueA(int s)*
- Kiểm tra queue rỗng: *isEmpty()*
- Đưa phần tử vào queue : *insert(x)*
- Lấy phần tử ra khỏi queue : *remove()*

## Các thao tác hỗ trợ trên queue

- Lấy giá trị đầu queue : *peekFront()*
- Kiểm tra queue đầy: *isFull()*
- Số phần tử trong queue : *size(x)*

# Cài đặt trên cơ sở mảng

Sử dụng

- Mảng **queArray[maxSize]** phần tử dữ liệu
- Chỉ số
  - **front** để chỉ đầu hàng đợi (nơi lấy ra)
  - **rear** để chỉ cuối hàng đợi (nơi đưa vào)

```
class QueueA {  
    private int    maxSize;  
    private int[]  queArray;  
    private int    front;  
    private int    rear;  
    private int    nItems;  
    //..... Các method .....  
}
```

# Các thao tác trên QueueA

```
//----- constructor -----  
public QueueA(int s) {           //  
    maxSize = s;  
    queArray = new int[maxSize];  
    front = 0;  
    rear = -1;  
    nItems = 0;  
}  
//----- put item at rear of queue -----  
public void insert(int x)  
{  
    if(rear == maxSize-1) rear = -1;  
    queArray[++rear] = x; // increment rear and insert  
    nItems++; // one more item  
}
```

# Các thao tác trên QueueA

*//----- take item from front of queue -----*

```
public int remove()
```

```
{
```

```
    int temp = queArray[front++]; // get value and incr front
```

```
    if(front == maxSize)
```

```
        front = 0;
```

```
    nItems--;
```

```
        // one less item
```

```
    return temp;
```

```
}
```

*//----- peek at front of queue -----*

```
public int peekFront()
```

```
{
```

```
    return queArray[front];
```

```
}
```

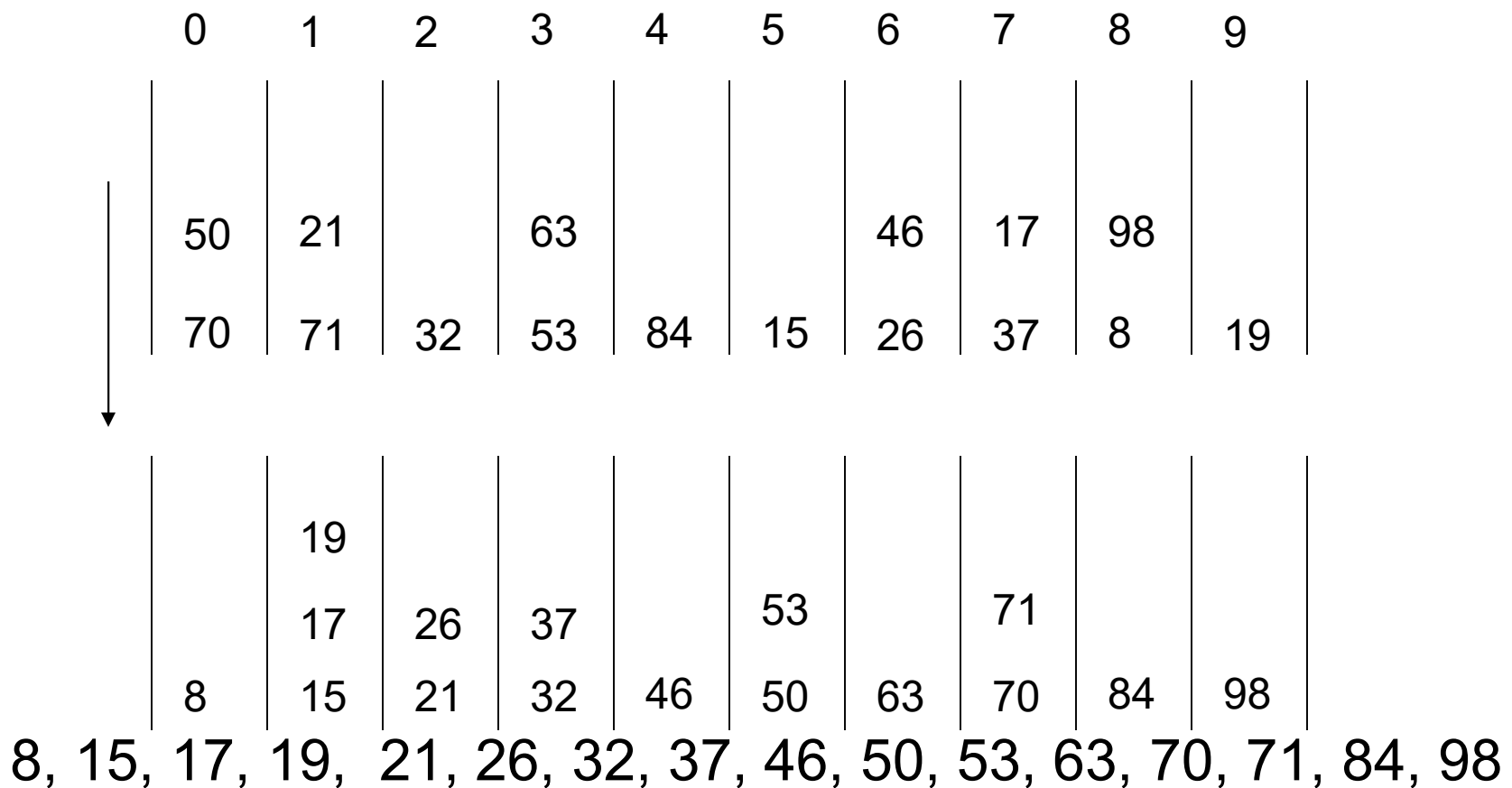
# Các thao tác trên QueueA

```
//----- true if queue is empty -----  
    public boolean isEmpty()  
    {  
        return (nltems==0);  
    }  
//----- true if queue is full -----  
    public boolean isFull()  
    {  
        return (nltems==maxSize);  
    }  
//----- number of items in queue -----  
    public int size()  
    {  
        return nltems;  
    }
```

# Ví dụ: Radix sort

- Sắp xếp dãy số nguyên dương tăng dần

71, 32, 53, 70, 50, 63, 84, 15, 26, 19, 8, 37, 17, 46, 98, 21





# Radix sort

*//Hàm phát sinh n số ngẫu nhiên*  
*static void ngaunhien(int[] a, int n)*

```
{  
    Random rd = new Random();  
    for (int i=0; i<n; i++)  
        a[i] = rd.nextInt(1000);  
}
```

```
static void xuat(int[] a, int n)  
{  
    for (int i=0; i<n; i++)  
        System.out.print(" "+a[i]);  
    System.out.println();  
}
```

*//Hàm đếm số chữ số*  
*static int scs(int n)*

```
{  
    int d = 0;  
    while (n>0)  
    {  
        d++;  
        n = n/10;  
    }  
    return d;  
}
```

# Radix sort

```
static void radixsort(int [] a, int n)
```

```
{  
    int i, k, m, lt, s, d, t, v;  
    QueueA[] Q;  
    Q = new QueueA[20];  
    for (i=0; i<20; i++)  
        Q[i] = new QueueA(100);  
    k = a[0];  
    for (i=0; i<n; i++)  
    {  
        if (k<a[i]) k = a[i];  
        Q[a[i]%10].insert(a[i]);  
    }  
    m = scs(k);  
    lt = 1;  
    s = 0;  
    d = 10;
```

```
    for (i=1; i<m; i++)
```

```
    {  
        lt = lt*10;  
        for (k=0; k<10; k++ )  
            while (!Q[s+k].isEmpty())  
            {  
                v = Q[s+k].remove();  
                Q[d+v/lt%10].insert(v);  
            }  
        t = s;  
        s = d;  
        d = t;  
    }  
    n = 0;  
    for (k=0; k<10; k++ )  
        while (!Q[s+k].isEmpty())  
            a[n++] = Q[s+k].remove();
```

```
}
```

# Radix sort

```
public static void main(String[] args)
{
    int[] a;
    int n;
    a = new int[100];
    Scanner x;
    System.out.print("Nhap so phan tu:");
    x = new Scanner(System.in);
    n = x.nextInt();
    ngaunhien(a, n);
    System.out.print("\nDay so ban dau:\n");
    xuat(a, n);
    radixsort(a, n);
    System.out.print("\nDay so da sap xep:\n");
    xuat(a, n);
}
```