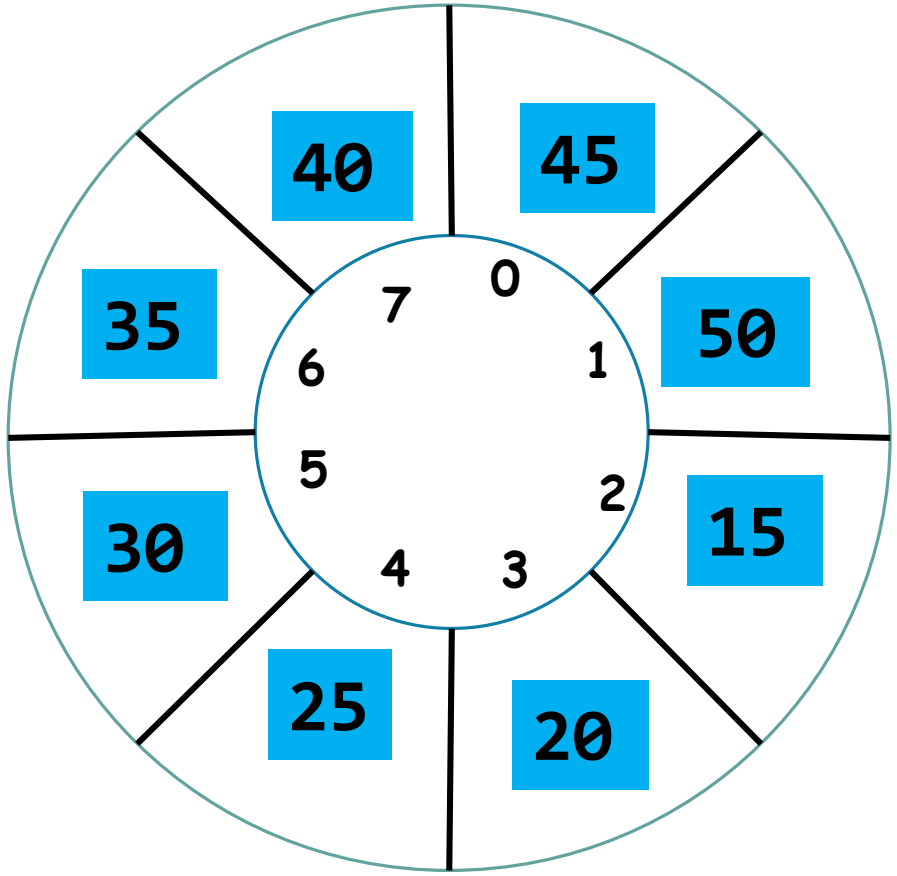
队列的ADT

```
ADT Queue is
  Operations
    Queue createEmptyQueue (void)
    创建一个空队列
    int isEmptyQueue (Queue qu)
     判断队列是否为空
    void enQueue ( Queue qu, DataType x)
     往队列中插入（或称推入）一个元素
    void deQueue ( Queue qu )
     从队列中删除（或称弹出）一个元素
    DataType frontQueue ( Queue qu )
    求队列头部元素的值
End ADT Queue
```

```
ADT Stack is
  Operations
    Stack createEmptyStack (void)
    创建一个空栈
    int isEmpty (Stack st)
     判断栈是否为空栈
    void push (Stack st, DataType x)
     往栈中插入（或称推入）一个元素
    void pop (Stack st)
     从栈中删除（或称弹出）一个元素
    DataType top (Stack st)
    求栈顶元素的值
End ADT Stack
```

3.9 循环队列



假溢出

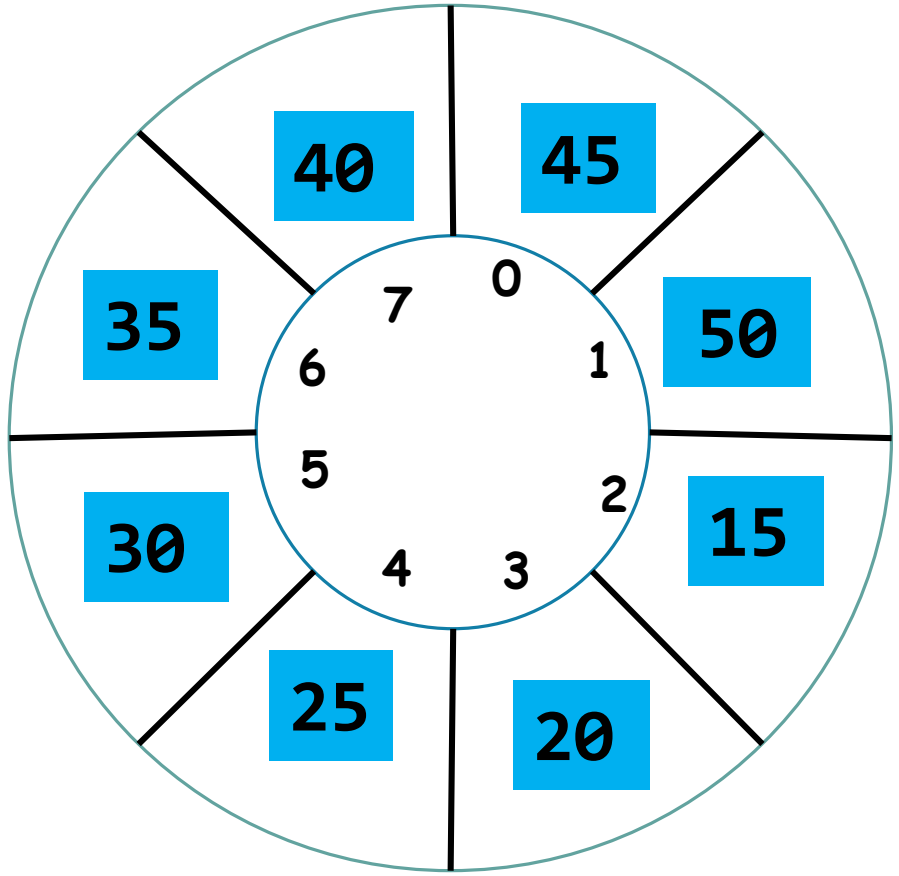
3.9 循环队列

假上溢：当前队列并不满，但不能入队

原因: 被删除元素的空间没有再被使用

解决: 环形队列（循环队列）

3.9 循环队列



入队：

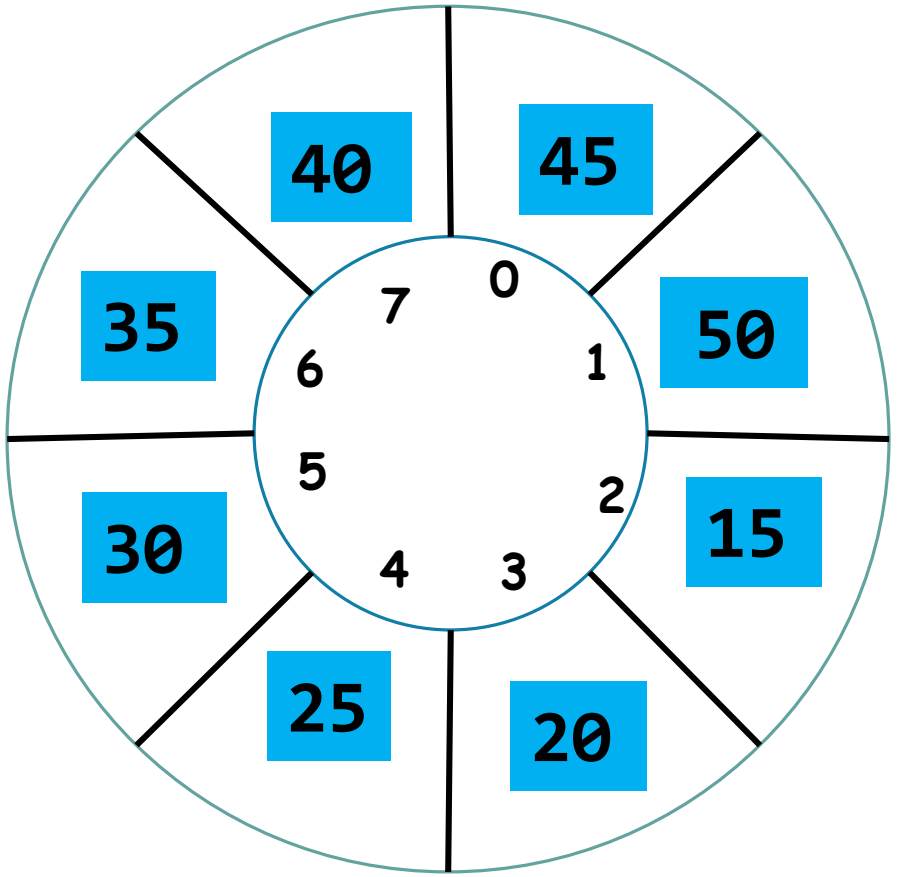
```
if   (squeue->r = MAXNUM)
        squeue->r = 0;
   else squeue->r ++;
```

利用模运算

```
squeue->r = (squeue->r +1) % MAXNUM
squeue->r = (squeue->r +1) mod MAXNUM
```

出队：

```
squeue->f =(squeue->f +1) % MAXNUM
```



思考： 判断队空和队满?

某一元素出队后，若头指针已从后面追上尾指针，
则当前队列为空：squeue->r == squeue->f

某一元素入队后，若尾指针已从后面追上头指针，
则当前队列为满： squeue->r == squeue->f

判断队列空的条件: squeue->r == squeue->f

判别队列满的条件： (squeue->r +1) % MAXNUM= =squeue->f

队列的顺序表示—假溢出

思考：只能用环形队列（循环队列）解决假溢出吗？

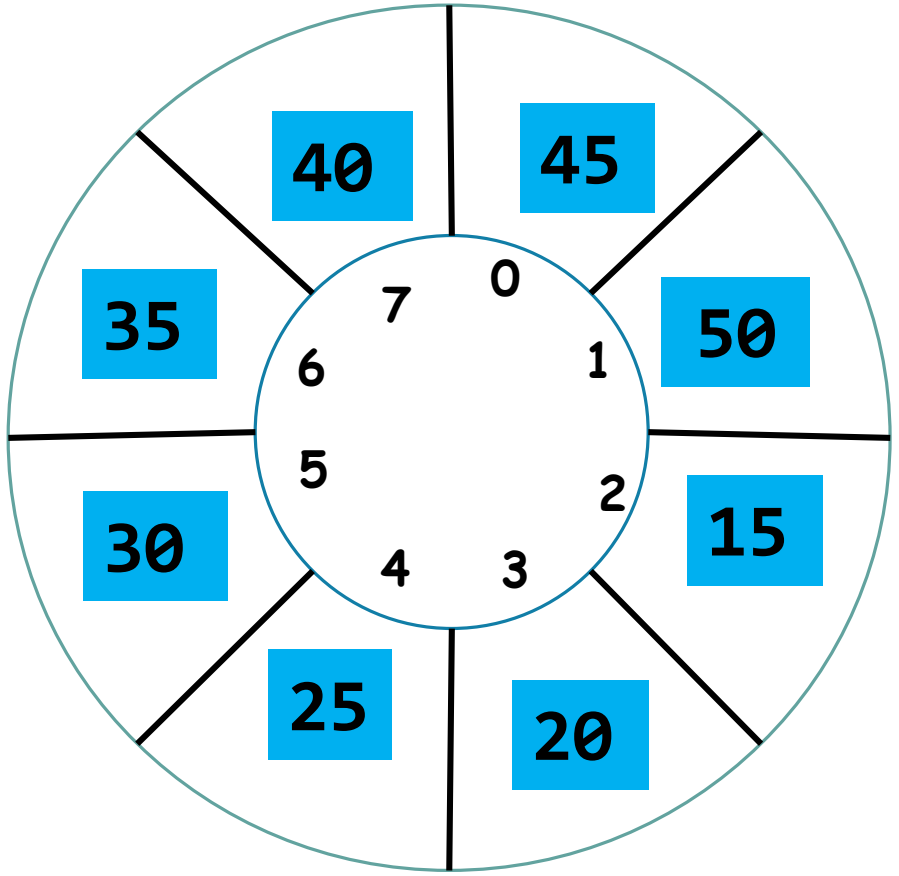
你还能想到什么方法呢？

3.9.3 入队　　算法3-21

算法思路：首先检查队列是否满，如果不满，则在队尾插入元素，并修改队尾指针。

```
1  void EnQueue_seq(SeqQueue squeue, DataType x)
2  {
3          if ((squeue->r + 1) % squeue->Max == squeue->f) //判断队列是否满
4                  printf("It is FULL Queue!");
5          else{
6                  squeue->elem[squeue->r] = x;   //元素x插入队尾
7                  squeue->r = (squeue->r + 1) % (squeue->Max); //队尾指针加1
8          }
9  }
```

3.9.4 出队



3.9.4 出队　　算法3-22

算法思路：首先检查队列是否为空，若队列非空，则删除队头元素，修改队头指针。

```
1  void DeQueue_seq(SeqQueue squeue)
2  {
3          if (IsNullQueue_seq(squeue))  //判断队列是否为空
4                  printf("It is empty queue!\n");
5          else
6                  squeue->f = (squeue->f + 1) % (squeue->Max); //队头指针加1
7  }
```