队列的学习

数组实现队列，使用两个指针——一个指向队头，一个指向队尾。

下标 0 1 2 3 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 40 | 50 |

|  |
| --- |
| 4 |

rear

front

|  |
| --- |
| 0 |

经过三次删除后

下标 0 1 2 3 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 40 | 50 |

rear

front

|  |
| --- |
| 3 |

|  |
| --- |
| 4 |

存在问题：空间未满，却因队尾无空间导致无法插入新元素，可利用循环数组解决。

0

1

|  |
| --- |
| 3 |

front

2

|  |
| --- |
| 4 |

4

50

rear

3

40

当尾部下标移除数组时，把它设为0，代码实现

rear = ( rear + 1 ) % QUEUE\_SIZE;

可将数组看组一个时钟，当下标刚移出数组时，如时钟般回到“0”，而下标以偏移量未2偏移数组最后一个元素（即下标值6），此时和数组下标1（有效下标）同一位置，一次类推。通过求余可得当前下标所标识的位置对应数组相应的有效下标。

存在的又一问题：循环数组为空或者为满时判断困难。

队列满时

60

70

50

80

40

3

2

2

3

4

1

0

front

rear

从队列中删除4个元素，front将增值4次，队列情况如图

80

2

2

2

3

4

1

0

front

rear

当最后一个元素被删除时

3

2

2

3

4

1

0

front

rear

可以发现队列为满或为空时，front和rear的值是一样的。

解决方法：一种为引入一个变量，用于记录队列中的元素数量，另一种数组中的一个元素始终保留不用。现采用后一种方法。

使rear指向插入的元素，front指向出队元素的后一个位置，front始终追赶rear，当队列为空时rear的值必须比front小1，而从队列中删除最后一个元素也恰好如此；当队列满时，因有一个元素未使用，所以此时rear的值恰好比front小2。当第一次或多次“环绕”到数组的头部时rear的值会大于数组的最大有效值，通过取余获得数组的有效下标。

队列为空满足的条件：（ rear + 1 ） % QUEUE\_SIZE == front

队列为“满”达成条件：（ rear + 2 ） % QUEUE\_SIZE == front