**实验三 基于Python实现循环队列**

目录

[一、循环队列简介 2](#_Toc116994469)

[1.1队列与循环队列 2](#_Toc116994470)

[1.2循环队列的代码实现 6](#_Toc116994471)

[二、Minst数据集 7](#_Toc116994472)

[2.1Minst数据集简介 7](#_Toc116994473)

[2.2Minst数据集代码实例 8](#_Toc116994474)

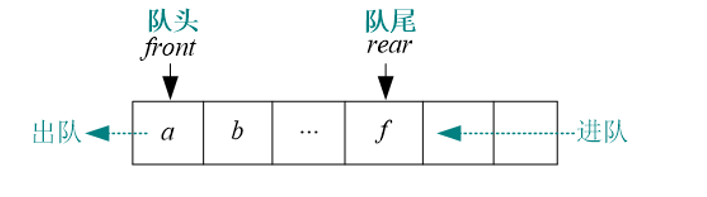
[三、python的time模块 8](#_Toc116994475)

[四、实验要求 9](#_Toc116994476)

## 一、循环队列简介

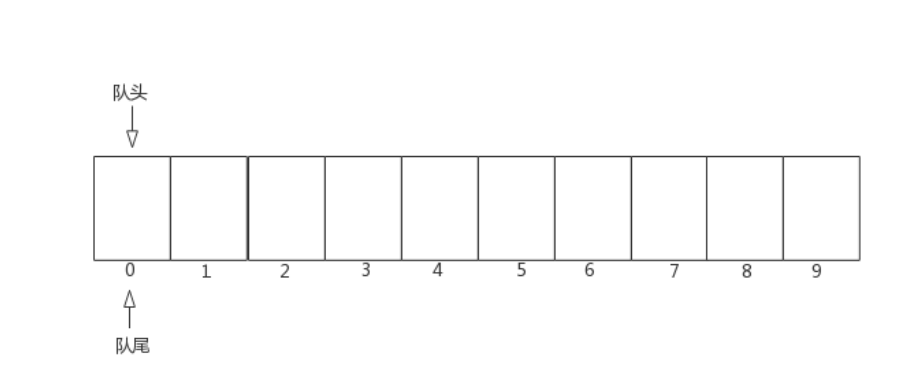
### 1.1队列与循环队列

先进先出的线性序列，称为队列，队列也是一种线性表，只不过它是操作受限的线性表，只能在两端操作。一端进，一端出。进的一端称为队尾，出的一端称为队头，队列可以用顺序存储也可以用链式存储。队列的顺序存储形式，可以用一段连续的空间存储数据元素，用两个整型变量记录队头和队尾元素的下标。



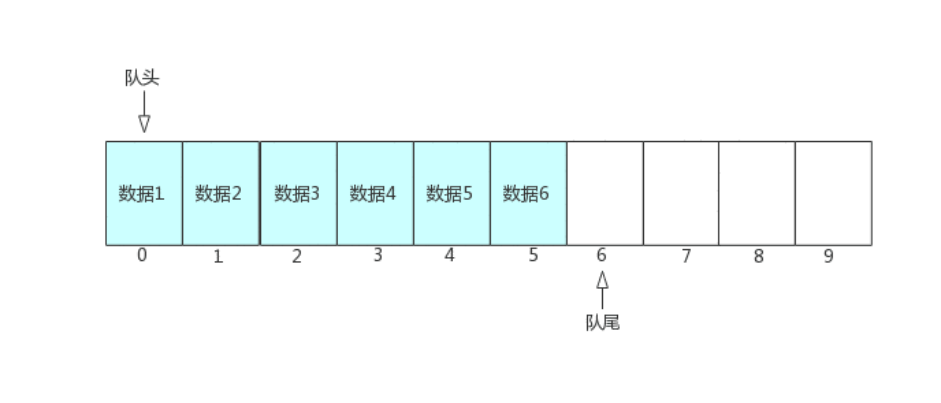
队列初始化

front,rear分别代表指向队头和队尾的"指针"(数组下标),构造空队列只需要按照所需队列长度申请一块内存给基地址,并且将队头指针与队尾指针赋值为0.



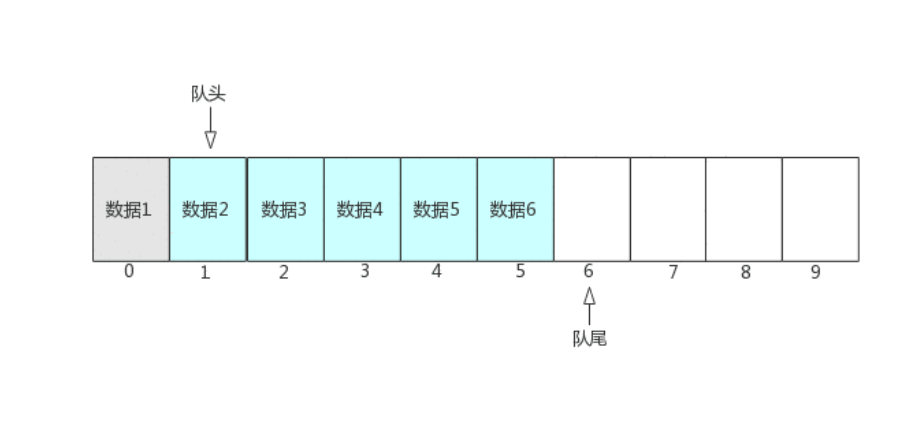
入队

将队尾指向的位置赋值,并且赋值后将队尾的位置后移

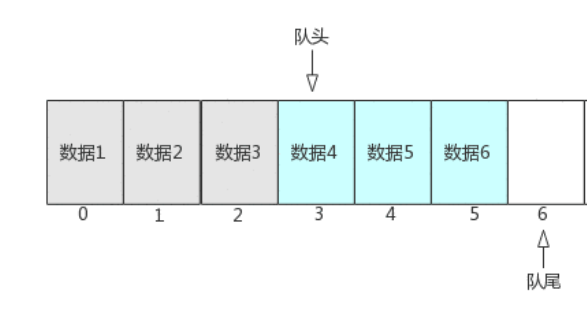


出队

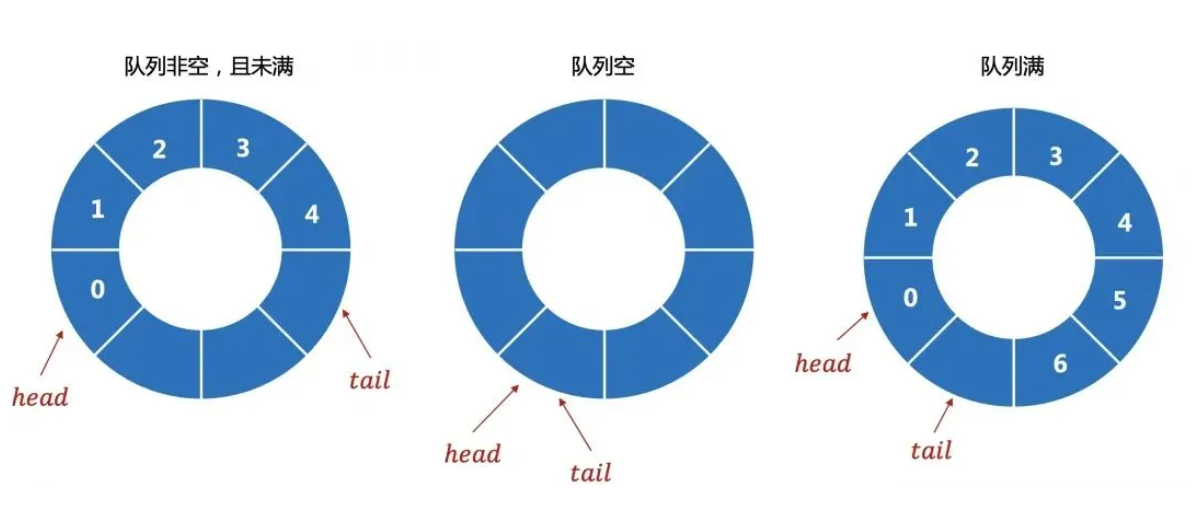
与入队类似，从队头指向的位置取值,取值后将队头的位置后移



普通的队列入队时会出现如下情况：



可以看出，此时rear已经指向末端，即队列已满，无法继续增加新数据，但申请的内存里，而队头前的数据早已读出，可以覆盖掉，为了利用存储空间，引入了**循环队列结构**



注意，循环队列的最后一位为预留位，是为了作为结束位，所以在创建循环队列申请内存时，创建长度应该为所用长度+1

### 1.2循环队列的代码实现



一个循环队列的类大致功能和框架如上，最主要的功能为入队和出队，入队需要判断队列是否已满，出队要判断是否队列为空，可以参考上图框架，也可以自己编写，符合循环队列性质即可。

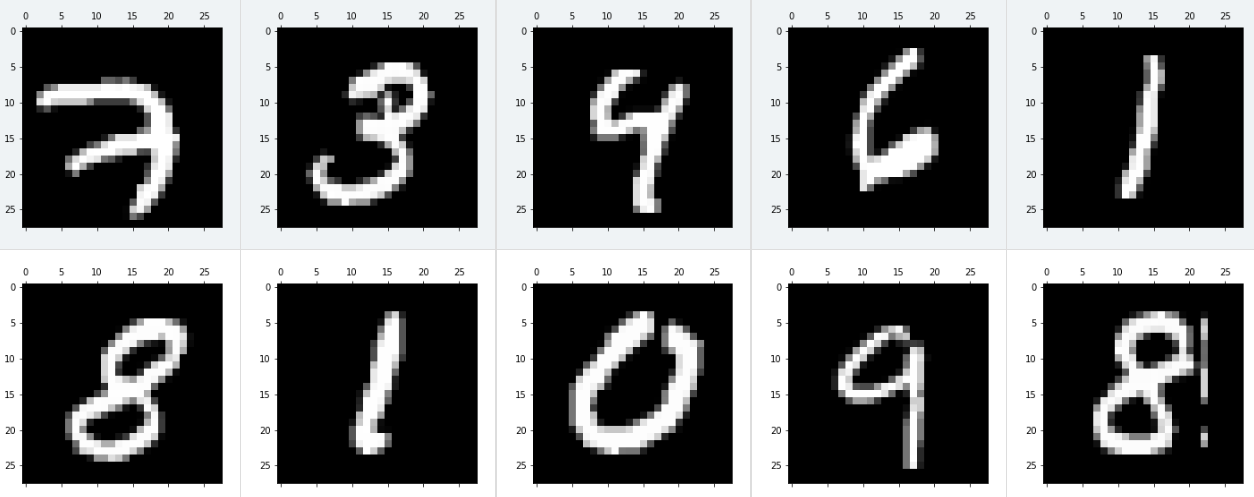
可以在网上检索相关循环队列的例子程序，请一定认真分析，在理解掌握的基础上借鉴使用。

## 二、Minst数据集

### 2.1Minst数据集简介

MNIST数据集来自美国国家标准与技术研究所, National Institute of Standards and Technology (NIST)。训练集（training set）由来自250个不同人手写的数字构成，其中50%是高中学生，50%来自人口普查局（the Census Bureau）的工作人员。测试集（test set）也是同样比例的手写数字数据，但保证了测试集和训练集的作者集不相交。

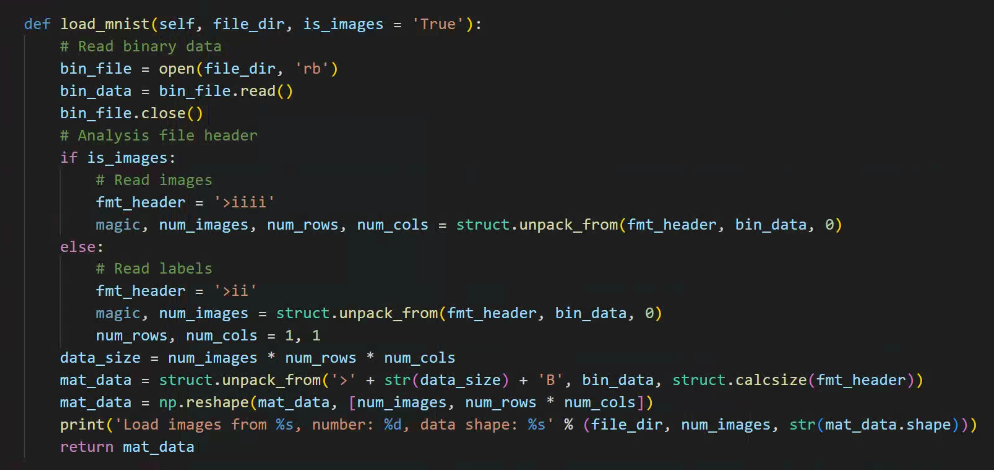
MNIST数据集一共有7万张图片，其中6万张是训练集，1万张是测试集。每张图片是28 × 28的0~9的手写数字图片组成。每个图片是黑底白字的形式，黑底用0表示，白字用0-1之间的浮点数表示，越接近1，颜色越白。



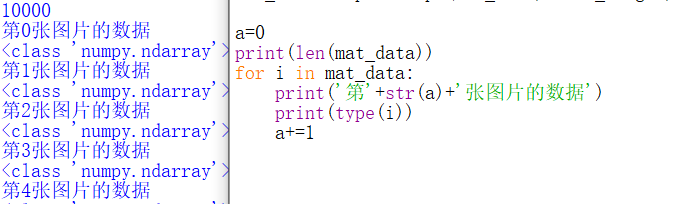
Minst手写数据集是以字节的形式进行存储，Minst的图像数据包前16个字节为4个整型int数据，涵义分别为，Magic帧，图像数量，行数，列数，其后的数据全为图像像素数据。即在前16个字节后，连续的数据都是28\*28=784为一幅图像的数据。

### 2.2Minst数据集代码实例

可以参考寒武纪的手写数字识别实验代码中的数据读取代码段来改写



该代码块的功能为取出所有的数据并存在mat\_data中，根据需要改写，比如我们本次实验只需要读取图片数据，并不需要读取标签数据。

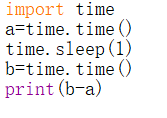


可以使用for循环对mat\_data数据遍历，注意此时i是一个np的矩阵，代表一张图片的像素信息，后续需要转换为Image格式或其他格式才能输出显示。

## 三、python的time模块

time库是Python中内置的处理时间的标准库，是最基础的时间处理库。

示例代码如下



代码输出



time.time()会返回从世界标准时间的1970年1月1日00：00：00开始到当前这一时刻为止的总秒数。

time.sleep()为延时函数，单位是秒，在实验中发现延时精度误差较大，并于cpu性能正相关，使用时应多注意。

## 四、实验要求

使用Python设计并实现可以存储Minst数据集中图像信息的类，从当前目录的data子目录读取数据集二进制文件minst\_images中每一幅图像的顺序编号（从0开始）和图像数据，存储到当前类对象对应数据成员，要求:

1. 编程实现从minst\_images文件中读取不少于1000幅图像样本，每个样本数据存储在题述类的对象里；
2. 设计一个循环队列存储题述类的对象，以4Hz频率向队列输入样本数据对应的类对象实例，队列需能够缓存不少于5秒时间的数据。同时，以2Hz频率从相应队列中取出图片并通过固定窗口顺序显示；
3. 当队列出现上溢出时，把队列读出频率提升8Hz，并体现为输出显示频率提升；
4. 当队列出现下溢出时，把队列读出频率降低回2Hz，并体现为输出显示频率下降。

以上流程，形成可持续运行的Python程序，关键代码片段和实验结果可以通过文本粘贴或截图体现在实验报告中。