



字符画大作业报告

高程大作业报告



设计思路与功能描述

Array

构造函数是通过 `template<typename ...Args>` 的参数包设计使得 `array` 类可以接受多个参数，然后通过获取参数包的信息来知道了这个数组的信息，例如维数多少，维数中每个储存数的大小，然后动态申请一个空间，还有一个很重要的就是要设计一个是否是子矩阵的判断。复制构造函数中就是要依靠是否是子矩阵来判断是否需要重新申请一个空间。然后是重要的 `[]` 的重载，我选择的方法是先在里面设计一个新的矩阵，然后把我需要的子矩阵的信息全部都复制过去，注意中间有一个判断，就是当子矩阵的维数小于 0，也就是比如原本是 `a[3][3]`，现在在找 `a[3][3][3]`，这种情况，就报错，然后退出，还有几个就是注意要记录好这个子矩阵的指针的位置，这样下一个子矩阵才能依照规律找到，然后还有就是一定要返回子矩阵，因为后续它的返回对象可能还是需要继续调用 `[]`，而 `[]` 之属于 `array` 类，然后注意到在返回的时候，会调用析构函数，将在函数里的 `array` 类析构掉，这时候就要注意了，因为它这时候指针指向的申请空间是原本的一维数组的内容，然后就是是否是子矩阵的判断就很重要了，在析构中，如果不是子矩阵就将那个对应的内容清除掉，如果是子矩阵就不 `delete`，这样就解决了这个问题，还有一个 `reshape` 函数，在通过 `template<typename ...Args>` 获取了新的数组信息之后，改变原本的 `array` 类里面的内容，注意到最后会统计 `reshape` 的里面的内容大小，如果和之前不一样，那就说明是不能够进行的，就报错，退出。还有重载 `=`，操作和复制构造函数基本相同，就是注意传入的值前面要加 `const`。还有给里面的申请空间赋值或者访问，就通过指针就好，然后就是要完成的矩阵加法除法等等重载，也很简单，就是构造出一个新的 `array`，然后让它等于其中一个传入的，这样就可以获得里面的维数等信息，然后将每项相乘，相除的结果放进去，注意如果一开始 2 个对象的维数等信息对不上，就不运算了，直接报错。

图片转化为字符画

首先是确定需要的字符，我选择的是作业推荐的，真的比自己找的要好啊啊啊，然后还有一个用来统计记录的字符，我也把他放在了全局函数里，因为考虑到比较大，放全局也比较方便。然后就是借助 `imread.readPic(photoName)` 和 `imread.getData` 来读取图片信息，注意到 `getData` 的参数直接换成了 `array` 类以及用来记录图片尺寸的 `int x,y`。进入主要增加的操作是完善 `array` 类的信息，比如然后录入每一个像素点的信息，我采取的是先按一维数组那样录入，然后就 `reshape` 成四维。第一个记录 `y` 坐标，第二个记录 `x` 坐标。然后获取完图片信息之后，就再开一个 `array` 类，专门用来记录灰度图的像素信息。采用的也是题目中推荐的计算方法。然后就是将记录的灰度信息转化为字符信息了，因为字符的长是宽的 2 倍，所以最后是以一个 `2*4` 的像素为单位，来转化为字符，读取这个 8 个像素点的总灰度，然后求平均，找到这个平均像素大小的对应的字符，然后储存好。完成后，建立 `FastPrinter printer` 这个对象，借助已有的 `testP_RandCF` 来快速绘制，设置背景全为白，前景为黑，然后就可以将这些字符绘制上去了。

视频转化为字符视频

因为最后没有优化到可以 40ms 内出一张图片，所以选择的还是先把视频转化成字符储存起来，再进行播放。

首先是利用 `pr` 将视频转化为图片，然后注意到输出的图片是有命名规律的，都是 `video1234`, `video1235` 这种，然后就依照这个从 0 开始一张一张的转化，具体的转化就和前面图片转字符画差不多。只不过没有了展示，为了表示进度，还设置了每完成 100 个就提示一下。

然后就是展示视频，主要的原理也就是每 40ms 展示一张图片，然后以此循环播放，先是，不过要注意这里是用 array 再次读取文件中的内容的，所以需要进行初始化，我找到的解决办法就是用 PicReader imread 先读取第 0000 张图片，将其中的长宽读出来，然后再初始化 array 类。然后就是从文件中读取信息到 array，再用 testP_RandCF () 展示出来。中间要设置好停顿时间。我采取的是每 40ms 放一张图片，所以就通过统计从一开始播放到现在的时间，然后和它应该的时间 $40 \times (i+1)$ 比较。然后停顿差的时间。顺便还用了 PlaySound 函数来进行了音乐背景播放。整体播放下来还是可以的。

在实验过程中遇到的问题及解决方法

在设计 array 类的时候遇到几个问题

当时第一个问题就是 array 中重载 [] 时，要返回临时变量，但是销毁临时变量的时候又会把它指向的保留有基础数据的底层给删掉，最后采取的办法就是加入一个 isSon 的变量来说明哪些是临时的子矩阵，哪些不是，然后在析构的时候就可以避免误删了。

然后当时因为知识点没掌握好，还出了一个奇怪的 bug，我用 new int (sum)，来申请空间，而不是 new int[sum]，然后就会出现有时候整个过程都没有问题，有时候又突然报错的结果，也就是随机 bug，后面找了好久才发现这个，我真的是 XXXXXX。

还有就是 Array& Array::operator=(const Array& a)的参数里面没有 const，导致的结果就是在用=赋值的时候找不到对应的函数，最后还是翻以前的练习发现的这个操作。

在转化过程中的问题

一开始有个严重的问题就是显示出来后总是断层，不连续，然后经过多个图片的尝试发现是因为需要绘制的字符画太大了，xy 过大，就会出这种 bug，然后发现它大概只能处理 y 是 200 以下的？反正就是比较小，但是后面经过 8 个像素点归为 1 个字符后，题目给的哪些测试图片都能过了。

然后就是字符太大了，不能展示出图片，然后我采取的方法就是运用 FastPrinter printer 来调整字体大小，最后发现还是 5 这种大小比较适合。

然后就是字符和像素对应的问题，这里一开始是没有相同为什么要用 2*4 的矩阵来整合像素的，后来自己实际画了一下才终于理解了。

在转化成视频的时候也有一个比较严重的 bug，因为一开始的 void testP_RandCF 中 FastPrinter printer 是函数里的临时变量，所以一次展示完了之后就会将控制台初始化，然后就会导致窗口一直在不断的缩小放大，然后一开始想到的方法是将 FastPrinter printer 变为全局函数，但是后面用多个函数并行的时候，会很不方便，最后改成了用引用传值，解决了问题。

还有就是将图片信息录入到文件中，一开始录入的时候就简单的用了 >> 来录入到文件中，然后发现对于一些图片，会出现录入到一半就结束的过程，猜测可能是录入过程中有停止 >> 录入的字符，然后改成了 put 解决了这个问题。

然后为了视频播放流畅和音频对上，我采用了实践课上分享推荐的方法，就计算到这一帧应该要过了多少时间，然后停顿时间差，这样可以保证整体就算有部分是慢了，整体后面还能跟上来。

心得体会

其实感觉这一次老师很手下留情了啊，留足了时间，工作量我感觉也少了好多，平时都要用五六天完成的大作业最后只用了 2 天就差不多了。主要是 array 那里卡壳了一些细节问题，果然基础知识很重要啊，不然总是会有些奇怪的 bug 出现。然后感觉这次往加分项视频那里过渡的也比较自然，最后比较可惜的是还是没找到办法能够把图片转化的时间控制在 40ms 内，差不多每张要五六百 ms，然后我去看了下那个时间分布，主要是 array 的操作比较慢，占了很多时间啊。希望后面能完善。

啊，高程就这么结束了，想想自己一年前还是对程序一无所知的小白，现在已经能写出一些程序了，还是很高兴很有成就感啊。特别是大作业感觉也特别锻炼人，通过它也提高了好多，也逐渐感受到了程序设计的奥妙，真的很开心能有这样的机会。

再见感谢各位老师，助教！我们有缘再见了 555555555555555555