实现分块检索

（采用Visual Studio 2017 15.7.2 Windows SDK 10.0.17134.0）

这可能是最后一次上机作业了，总的来说还算是比较顺利的。通过文件输入，读入第一次创建合适的存储空间，然后再读入一次把各个块各个位置的数据录入，通过折半查找条目第一个索引，然后再在一个条目内顺序查找，找到条目或者返回值说明不存在。

0、程序要完成的内容

文件输入数据，实现块间采用折半检索，块内采用顺序检索的分块检索程序。

1、最终程序的结构是这样的：

首先提示用户指定输入的文件（稍加修改也可以手动输入），然后逐行扫描文件，确定区块数（行数），总共条目的数量（整型数组大小），每个区块的起始下标和长度，把它们保存起来（区块索引为索引节点数组，每个节点包含起始位置下标和区块长度（数据保存在一个完整的常数组里面，通过重载区块索引节点的 ”int& operator[](size\_t i)” 运算符达到通过两个下标访问数据的操作））。

然后就安照用户命令进行查找等操作就可以了。算法就是老师讲的算法。

2、一些遇到的问题和解决方法

实际上在一开始写折半查找的算法时，用的不是首，尾，中间的三个变量进行夹逼的方法，而是真的用总长度除以二，然后每次再除以二为每次的步进长度进行比较，然后确定步进的方向一步步进行查找。

这个算法看起来只需要一个参变量就能解决问题，但实际上是有问题的。如果折半查找的全体的索引的数量为8，索引依次为0，1，2，3，4，5，6，7 。如果我要查找索引为7的条目，首先会找到3，然后5，然后6，然后就没了，查找失败，返回不存在（然而结构是错的）。

这是整数的性质决定的，最开始想的解决方案是步进长度+1，然后就会出现步进超过范围的问题，然后把超过范围定义为达到最末端，也不一定能保证一定能再找到之前要找的内容。这样的查找只是比较适合2^n-1长度的查找。或许可以直接用比总条目数n大的最小2的次方作为“虚拟的”总长度，然后进行这样的折半查找（注意超过范围的中间数的处理）。这样的话其实仍可以保证最多步进次数最小（当然，不然也就不叫折半查找了）。

不过这样的的确确不如直接用三个变量显性地控制查找范围和当前位置，虽然多使用内存（也没多多少），但代码明显更好维护，更直观。最终采用的也是这样的算法。

最开始并没有注意到可以要求各个区块的大小相同，可以提前要求用户输入总条目数和每个块的大小，所以使用了下标的表示，而不是采用二维数组的形式（c++ 中直接进行二维数组的动态分配会要求每行的元素个数一样，正好可以用到输入的数据块大小的数据直接构造）。

实际运行中，因为没有对不同长度行数的数组进行查找（因为发现没有需求），所以在这个最开始设计的时候考虑了并专门设置的功能并没有很好地进行调试。所以可能运行并不是很稳定，也算是有一些遗憾。

3、一些可以改进的地方

首先应该就是手动输入的支持可以再考虑支持一下。然后就是关于不定长的区块的调试以后还是要慢慢处理。以后的话可以考虑把这些这学期写的算法和数据结构搞到一起，形成我的一个库。反正我所有的功能基本都是用类来封装的，再用命名空间（namespace）加以区分，应该还是不错的。