算法

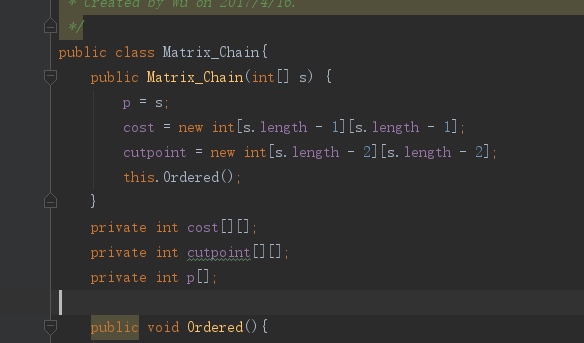
Pratice 2

14130130231

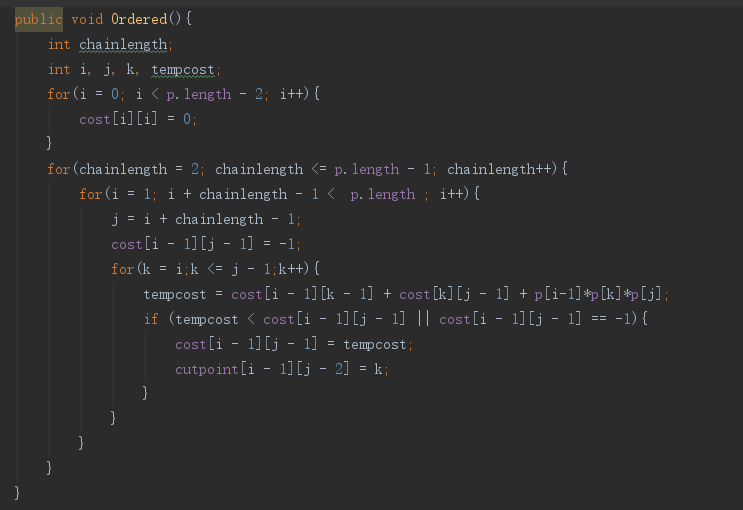
吴天成

1. ***Matrix-chain product. The following are some instances***
   1. ***<3, 5, 2, 1,10>***
   2. ***<2, 7, 3, 6, 10>***
   3. ***<10, 3, 15, 12, 7, 2>***
   4. ***<7, 2, 4, 15, 20, 5>***

最优矩阵链乘法,对于输入的矩阵大小数组，求出最优乘积方案



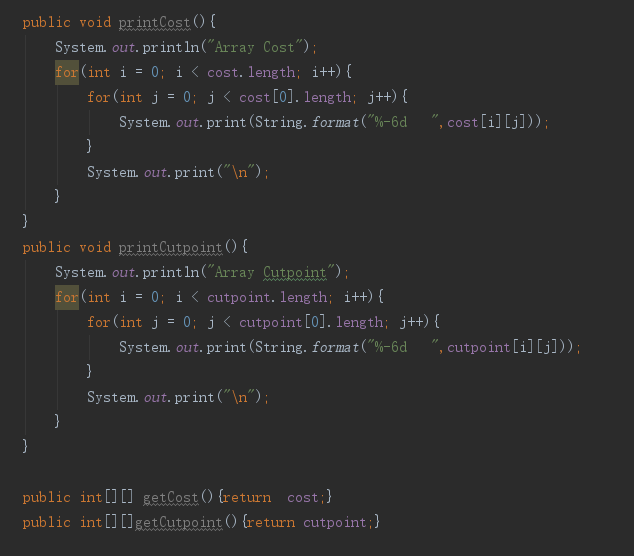
Matrix类中含有私有私有变量 cost二维数组用于保存数组相乘的代价以及cutpoint二维数组用于保存矩阵链相乘的切割点。Ordered方法用于求出对一条矩阵链相乘的最优代价。代码如下

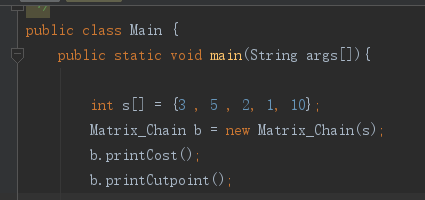


Ordered方法中，chainlength变量表示链长，i变量表示矩阵链的初始矩阵，j表示矩阵链的最后一个矩阵，k变量表示相乘时的最优切割点。

此方法首先初始化一个矩阵相乘代价为0，再从链长为2，从第一个矩阵开始，两个矩阵相乘开始动态计算最优代价。在最内层循环中计算最优代价时每次都使用了保存在cost数组里子问题的最优解。

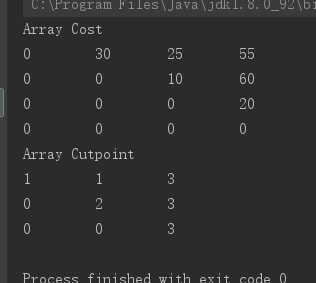
Matrix类中还有其他方法printCost(),printCutPoint,getCost(),getCutPoint用于对私有变量的获得与输出。代码如下

  
main函数代码如下

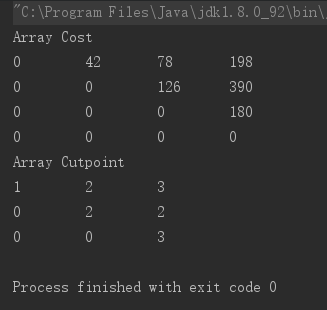


对于题中所给数据带入main函数可以得到以下结果

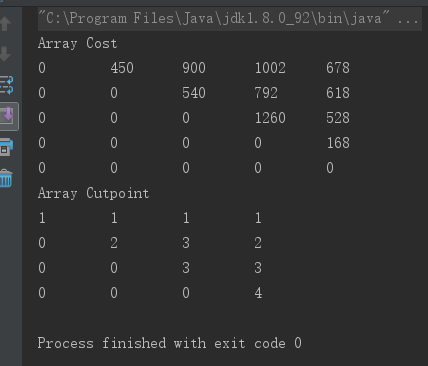
1. <3, 5, 2, 1,10>



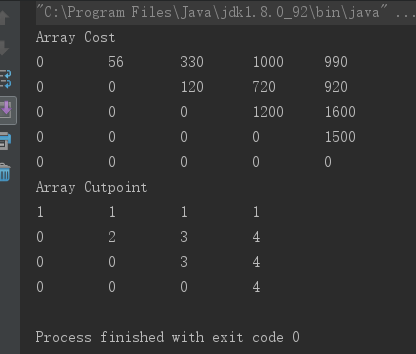
1. <2, 7, 3, 6, 10>



1. <10, 3, 15, 12, 7, 2>



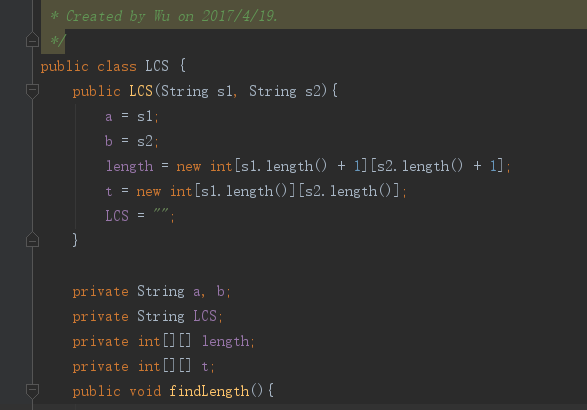
1. <7, 2, 4, 15, 20, 5>



1. **Longest Common Subsequence (LCS). The following are some instances.**
   1. **X: *xzyzzyx* Y: *zxyyzxz***
   2. **X: *ALLAAQANKESSSESFISRLLAIVAD***

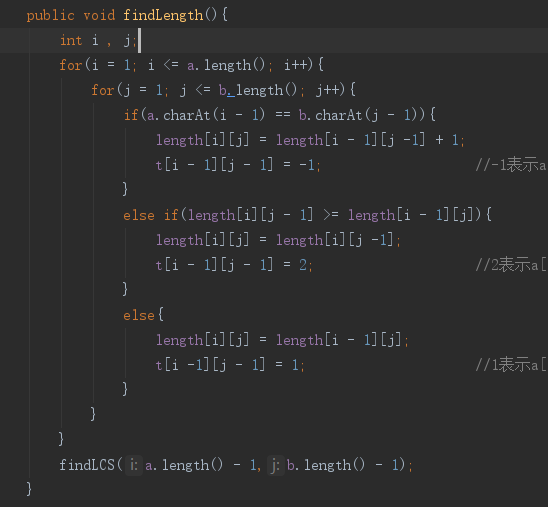
**Y: *KLQKKLAETEKRCTLLAAQANKENSNESFISRLLAIVAG***

找出两个字符串的最长子序列,类LCS如下图所示



方法findLength为找出最长子序列的元素和长度。二维数组length用来保存最大子序列的长度

代码如下



方法中省略了对length数组行与列任意一个数为0的情况下数组元素值为0的初始化，因为java中new出一个数组后自动对数组进行的赋0 的初始化。

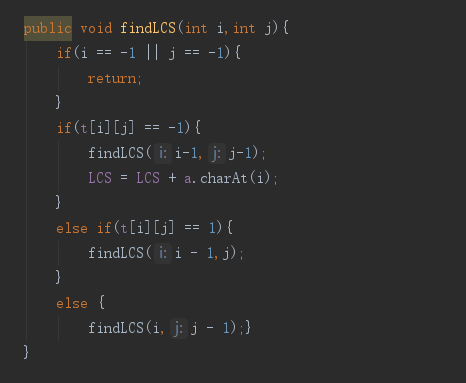
Length[i][j]表示第一个字符串包含0-i的子串与第二个字符串0-j的子串的最长子序列的长度

t[i][j]不同的值表示length[i][j]的值是从哪个操作得到的

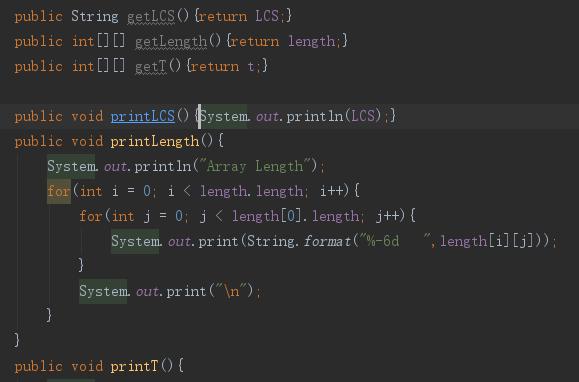
该方法自底向下开始，从第一个字符串第i(i = 1)个元素起与第二个字符串第j(j = 1)个元素进行比较，若相等，则此元素是两个字符串的子序列之一,对length对应的元素+1，对辅助数组t[i-1][j-1]赋值。若不相等，则比较length数组[i-1][j] 与[i][j - 1] 的大小，对辅助数组t赋值。循环求解可求出。

循环结束后有一个方法为findLCS，用于通过两个数组得到LCS值

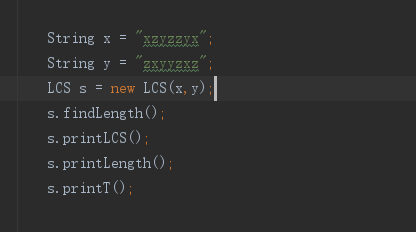
代码如下



类中还有其他方法用于对获取私有变量值以及输出操作

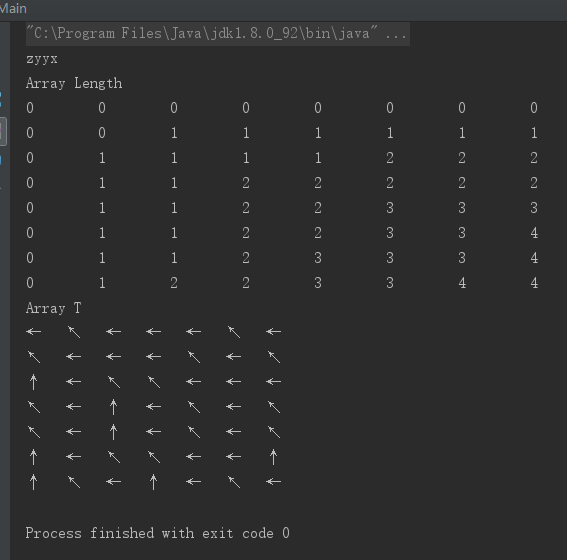


Main函数代码如下



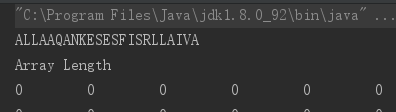
将题中数据带入main函数可得结果

a) X: xzyzzyx Y: zxyyzxz



b) X: ALLAAQANKESSSESFISRLLAIVAD

Y: KLQKKLAETEKRCTLLAAQANKENSNESFISRLLAIVAG

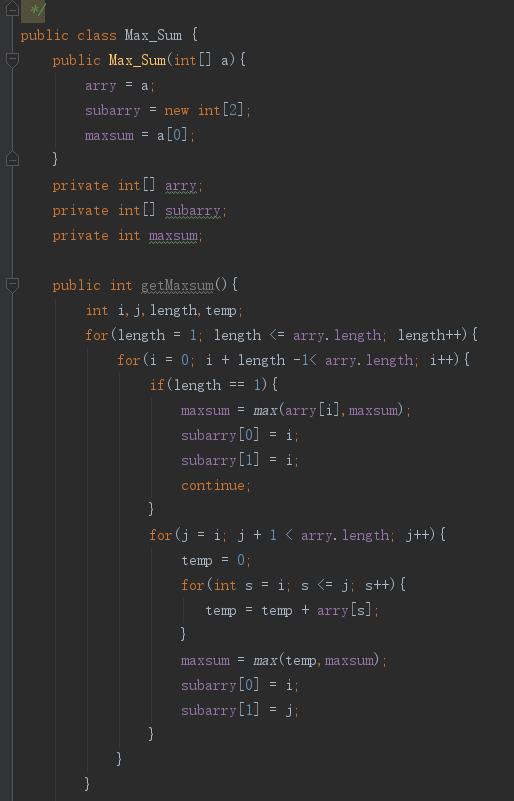


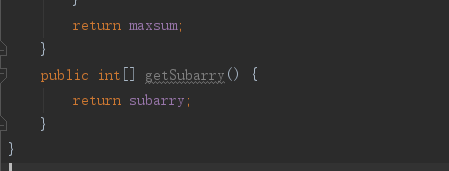
由于数组过大，所以只截最长子序列的结果

1. ***Max Sum. The following are some instances:***
   1. ***(-2，11，-4，13，-5，-2)***

求数组的有对大相加和的子数组

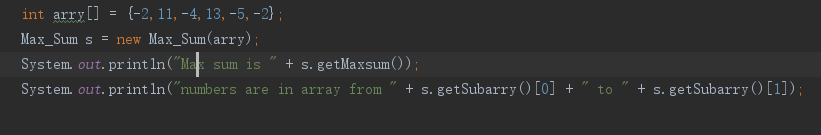
类Max\_Sum如下图所示



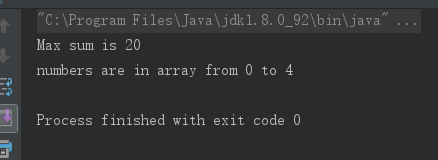


其中getMaxsum方法可以得到最大子数组的求和值，其思路与第一题相类似，length为子数组的长度，i表示子数组的首位下标，j为子数组的末尾下标。通过从子数组长度为2开始循环，计算出子数组相加的最大值。由于是相加操作，不像第一题中的相乘操作使时间复杂度递增，所以没有采用辅助数组来记录不同子数组之间的最大相加值。

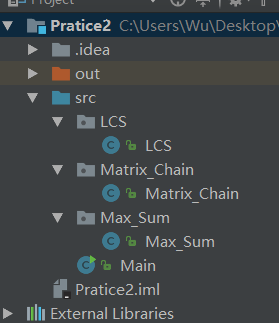
Main函数代码如下



可以求出结果为



工程结构如下



代码在Pratice2/src中