**栅格外扩法**

**一．输入图层**

road\_test.shp：提供的大学线路，共有4个等级1,2,3,4（weight属性）。

test\_pt.shp：测试的点集。

**二．流程**

1.由原始矢量图层road\_test.shp提取某一等级线路，生成栅格数据为road\_2。（只能提取某一等级道路作为种子扩充是有缺陷的，假设提取得道路的最高等级为2，因为实际上比等级2更低的道路也对于区域的扩充有影响，因此这样的提取得到的区域会小于等于实际上应该得到的区域，这也是使用栅格数据的一个缺陷。）

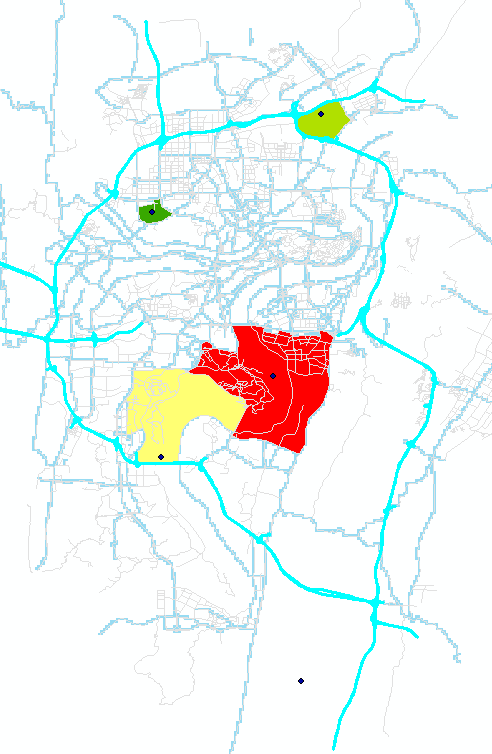
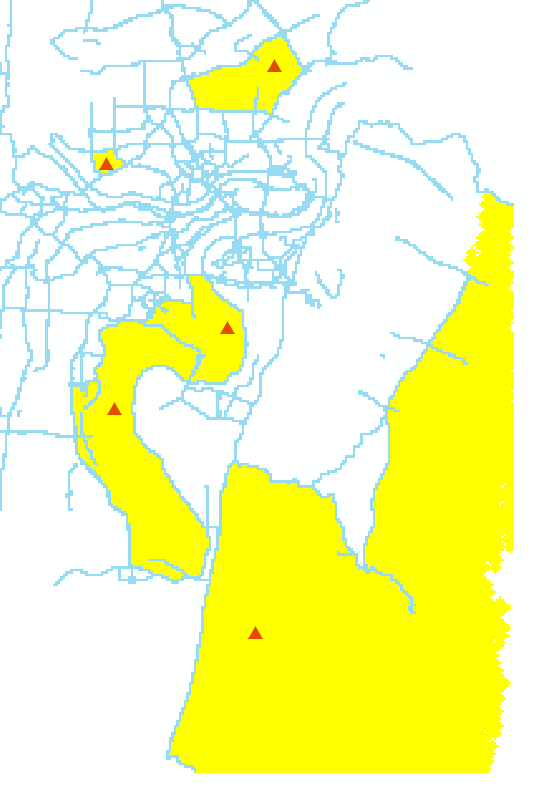
2.提取输入测试点图层的点的坐标，计算得到该坐标在栅格数据中的行列号，并以此作为种子生长点。

3.以4-邻域为搜索范围，以步骤2中的种子点进行种子搜索。搜索的终止条件为搜索到了边界或者是相应等级的道路。测试点集test\_pt.shp有多个位置不同的点，每个种子点的搜索范围不同，输出不同的tif。

**三．结果与分析**

栅格外扩法结果图如下左图所示。该测试点集中共包含5个点。其中蓝色为等级2的道路栅格数据，红色三角为种子点。黄色栅格部分为根据生长算法得到的种子点相应的提取区域。

利用种子生长法的一个劣势在于其需要不断的压栈，这也是导致程序不稳定的原因之一。观察最右下角的测试点，该测试点位于道路围合区域之外（实际上，该点所形成的区域不应该被提取出来，这也是使用栅格数据的缺陷之一）。由于栅格数据无法判断测试点是否落入围合多边形内，因此右下角的点会不断生长，由于数据过大，此时会使得程序因为递归太深（估计递归太深，压栈过多，而真实数据量并没有多少，大约1.5万多个格网左右）而自动结束。为了使程序能够进行，因此我勉强想了对策，由于无法判断递归有多深，因此近似采用由种子生长点的单位个数作为一个限制，当生长超过一定限制的单元数目时，则停止该种子点的生长。因此右下角的结果便是限制的结果。



矢量外扩法如上右图所示。其中，最底层浅灰为实际道路网，浅蓝色为设置权重为2的道路，高亮蓝色为权重为1的道路（即等级最高的道路）。四点提取区域以不同颜色表示。可以对比发现，矢量外扩法较栅格的优势在于：

1. 可以充分考虑不同等级道路的影响。矢量外扩法考虑多个等级线路得到结果，而栅格仅考虑一个道路等级（若要栅格考虑多个等级，实际在处理过程中情况复杂，难以判断进行种子生长）。

2. 得到的结果更符合实际情况（如右下角位于包围区域的点没有被提取），而栅格数据精度受分辨率影响。

3. 栅格法的计算受栅格数据量的影响。

栅格外扩法较矢量的优势在于：

1. 流程简单，算法清晰。

2. 速度较快。限制矢量外扩法的计算瓶颈在于，对于要素存储为shp图层时进行的要素插入步骤，也就是说插入的元素越多（不知道arcpy是否存在批量插入的函数且批量插入的效率会高），那么速度就越慢。就算法本身计算而言，效率也是比较高的。