

## Problem 2 - Merging Clusters

 [courses.edx.org/courses/course-v1:MITx+6.00.2x\\_4+3T2015/courseware/061b1b4da2fd4a8db1cb9b5d7db39208/96c6b9a4b9894](https://courses.edx.org/courses/course-v1:MITx+6.00.2x_4+3T2015/courseware/061b1b4da2fd4a8db1cb9b5d7db39208/96c6b9a4b9894)

Below is a simpler datafile (test.txt). As with cityTemps.txt, the first line represents the name of a point. The lines after that, up until before #end represent how many features will correspond to each point and they are represented by numbers on the same line as the point name, comma delimited. The line #end represents the end of the column titles and the beginning of the datapoints. You must have an empty line at the end of the file.

```
#point_name
#feature_value1
#feature_value2
#end
a,3,1
b,6,2
c,6,5
d,6,2
e,5,5
f,1,4
g,5,8
```

Appropriate test lines with this data would be:

```
points = buildCityPoints('test.txt', False)
hCluster(points, Cluster.singleLinkageDist, 3, False)
hCluster(points, Cluster.maxLinkageDist, 3, False)
hCluster(points, Cluster.averageLinkageDist, 3, False)
```

And with a correctly implemented set of functions, one possible output is shown below. For such a small number of data points and features it is possible that your output would be slightly different, depending on which point gets chosen in case of a tie.

Final set of clusters:

```
C0:a
C1:b, c, d, e, g
C2:f
```

Final set of clusters:

```
C0:a, b, d
C1:c, e, g
C2:f
```

Final set of clusters:

```
C0:a, f
C1:b, c, d, e
C2:g
```

