Problem 2 - Merging Clusters

In this problem, you will finish implementing the `ClusterSet` class by writing code for the three missing functions: `mergeClusters`, `findClosest`, and `mergeOne`.

- `mergeClusters` will create a new cluster containing the union of the points in c1 and points in c2. This new cluster will be added to the cluster set, while c1 and c2 are removed from the cluster set. This funcion does not return anything.
- `findClosest` will use the "linkage" parameter to find the distance between two clusters. It will iterate over all pairs of clusters in the cluster set and return the tuple (c1,c2) of the clusters within the cluster set that are closest. Note that no matter what linkage criteria we are using, we will always return the cluster pairs that are closest to each other.
- `mergeOne` will make use of `findClosest` to determine which pairs of clusters to merge. Then, it will use `mergeClusters` to perform the merging on these two closest clusters. This function returns the tuple (c1,c2) representing the clusters that were merged.

To test how your code clusters the city data, you may use the `hCluster` function and uncomment the line `#test()` to run the hierarchical clustering algorithm. It may take up to a minute to cluster, so be patient. Notice that the last parameter of `hCluster` is a history flag. If toggled, it will print out more detail, in particular which clusters are merged at each step. During testing, you may also want to make up a new datafile that contains less datapoints, less features, and easier numbers to work with.

Hint: A simpler datafile and sample output

Enter all code for the `ClusterSet` class below, including the functions in this class that were already defined for you. Do not paste the `Cluster` class code.

Test: cluster1

```
Testing simple datafile with mergeClusters and no scaling
```

Output:

```
       C0:a, b, c, d
```

Test: cluster2

```
Testing simple datafile with findClosest and no scaling
```

Output:

```
    a[2.0], b[2.0]  and  d[4.0], e[2.0]
    a[2.0], b[2.0]  and  c[1.0], f[3.0]
    a[2.0], b[2.0]  and  d[4.0], e[2.0]
```

Test: cluster3

```
Testing simple datafile with mergeOne and no scaling
```

Output:

```
    a[2.0], b[2.0]  and  d[4.0], e[2.0]
```