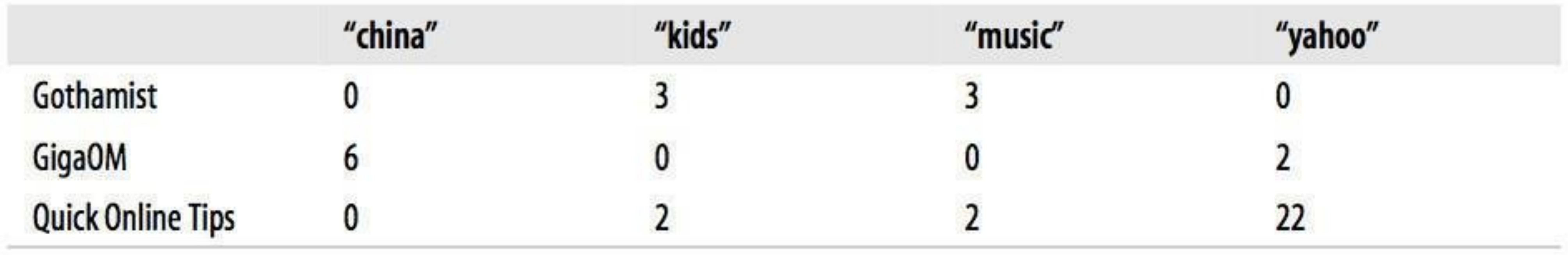
ENGR 102 PROGRAMMING PRACTICE

WEEK 9

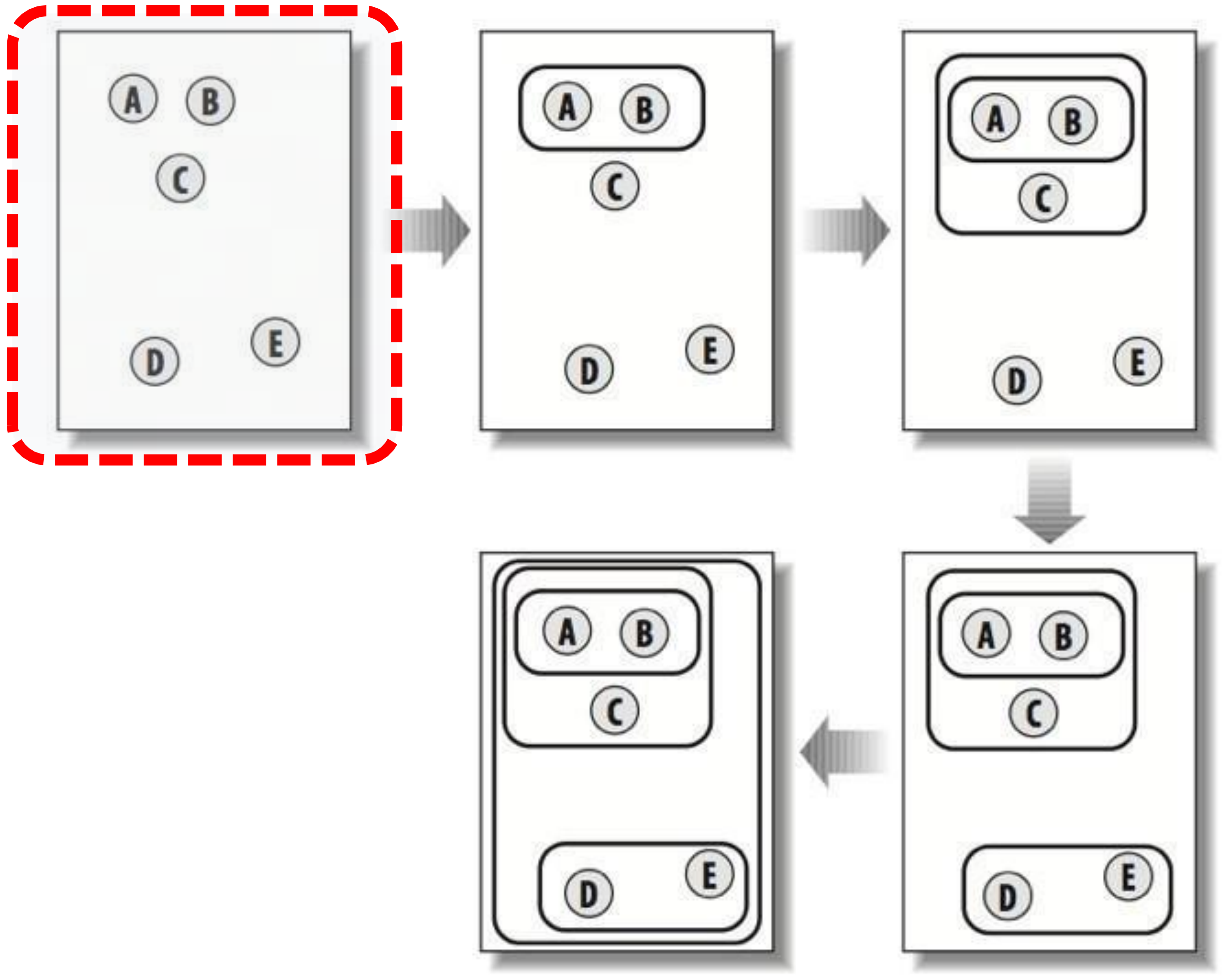
Structuring & Visualization

Example

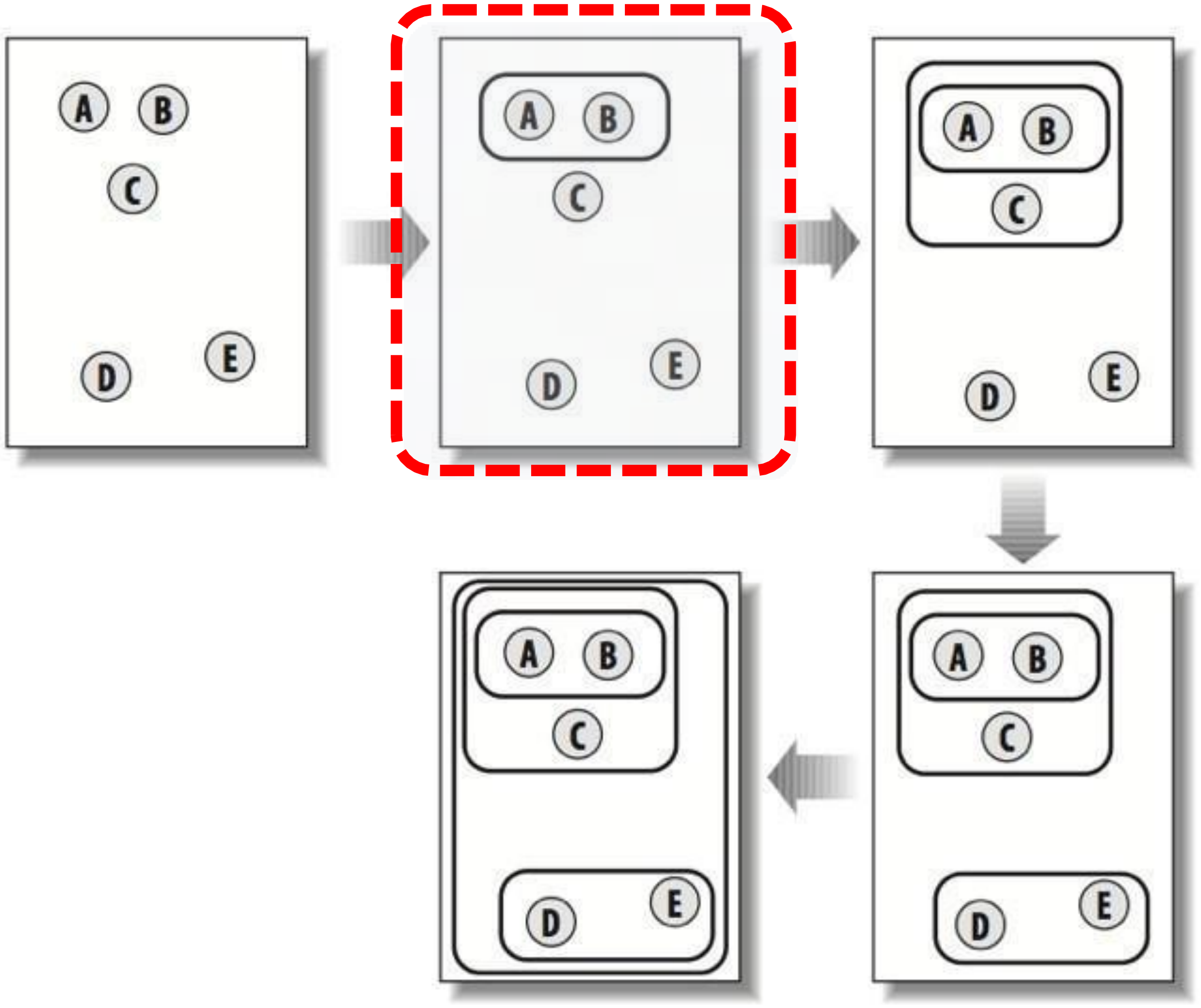


	"china"	"kids"	"music"	"yahoo"
Gothamist	0	3	3	0
GigaOM	6	0	0	2
Quick Online Tips	0	2	2	22

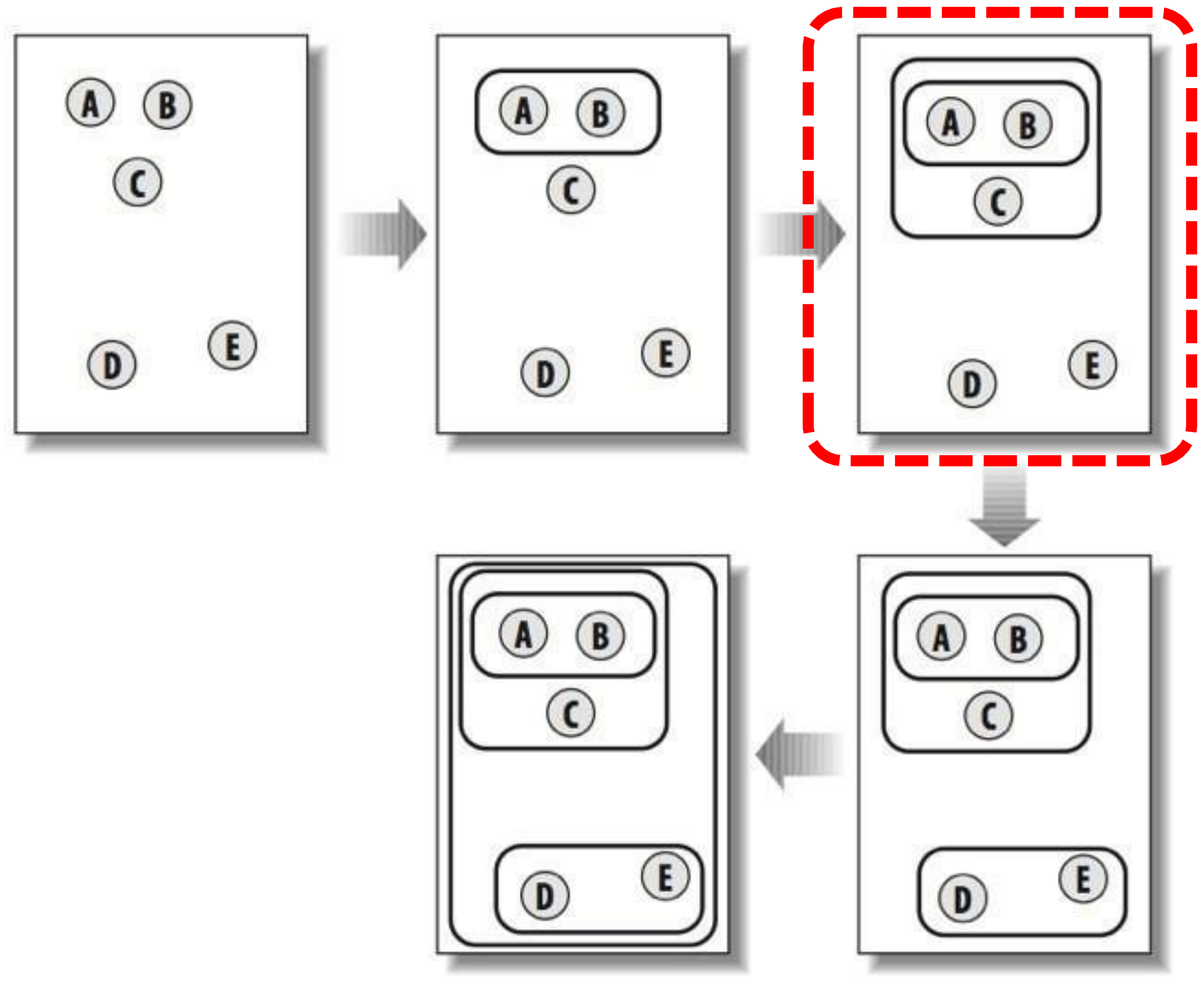
Hierarchical Clustering



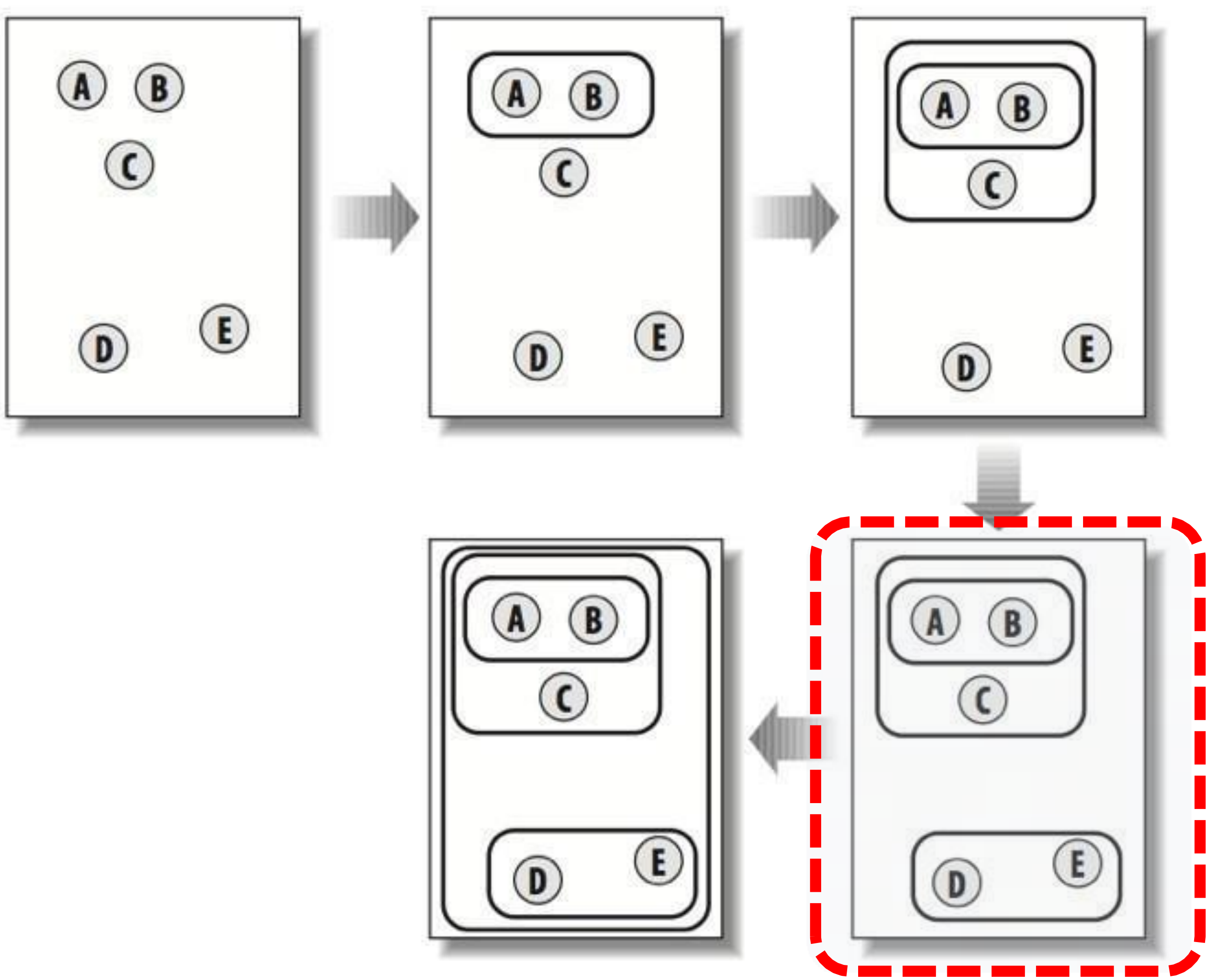
Hierarchical Clustering



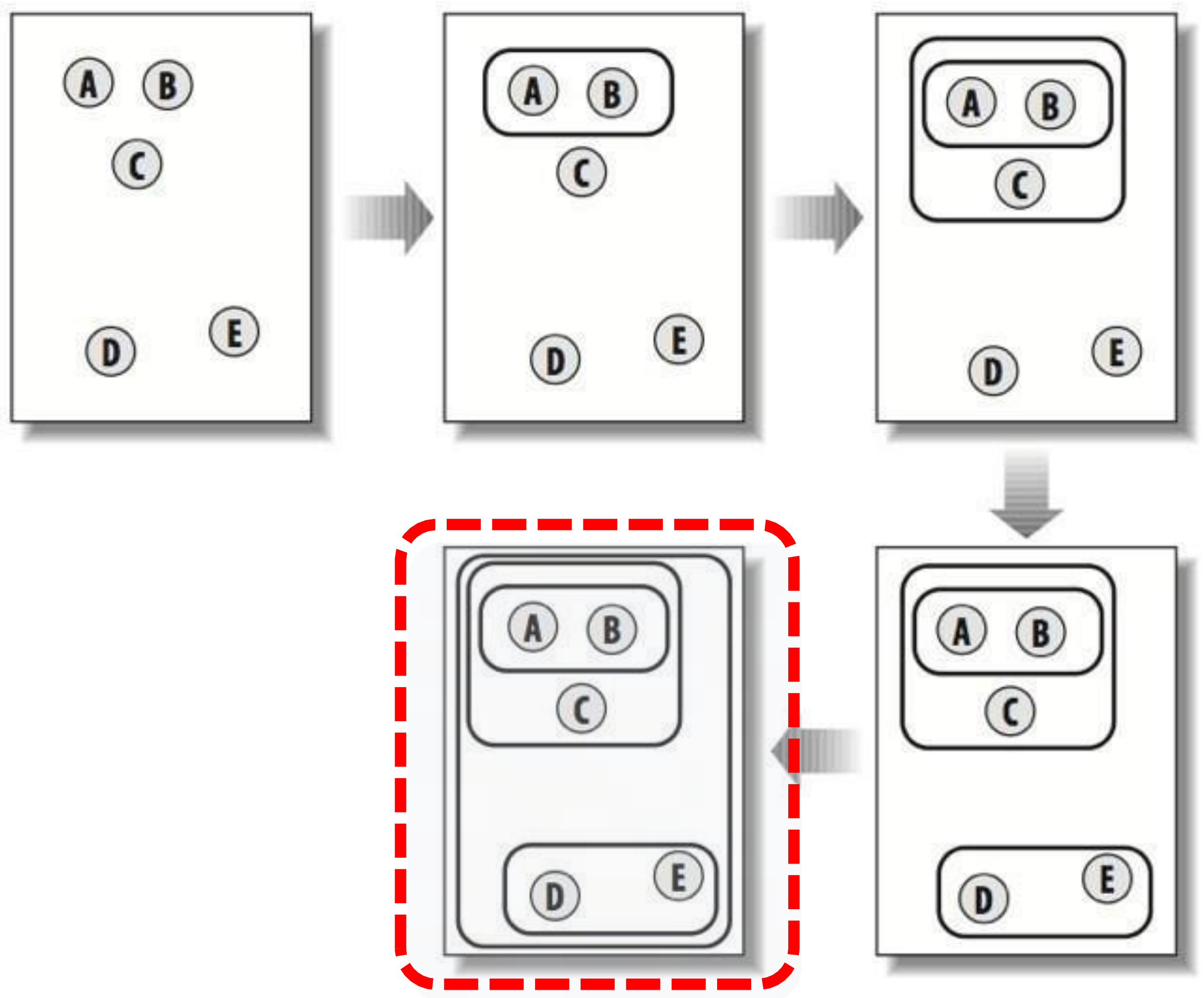
Hierarchical Clustering



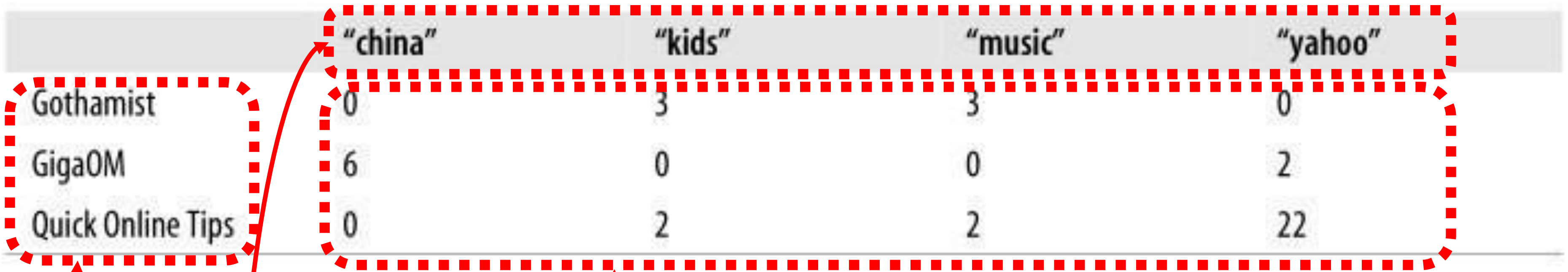
Hierarchical Clustering



Hierarchical Clustering



Play with H. Clustering

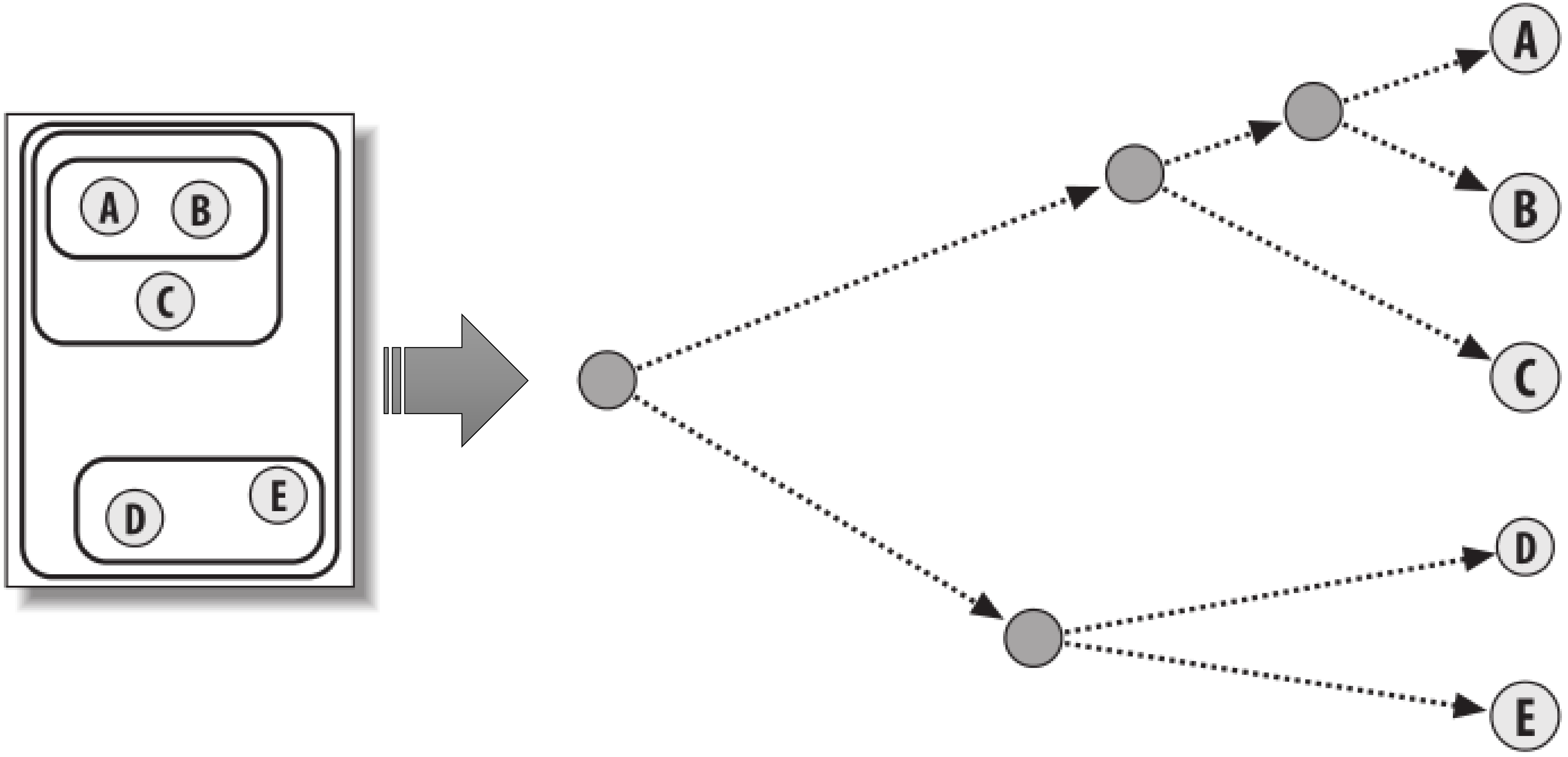


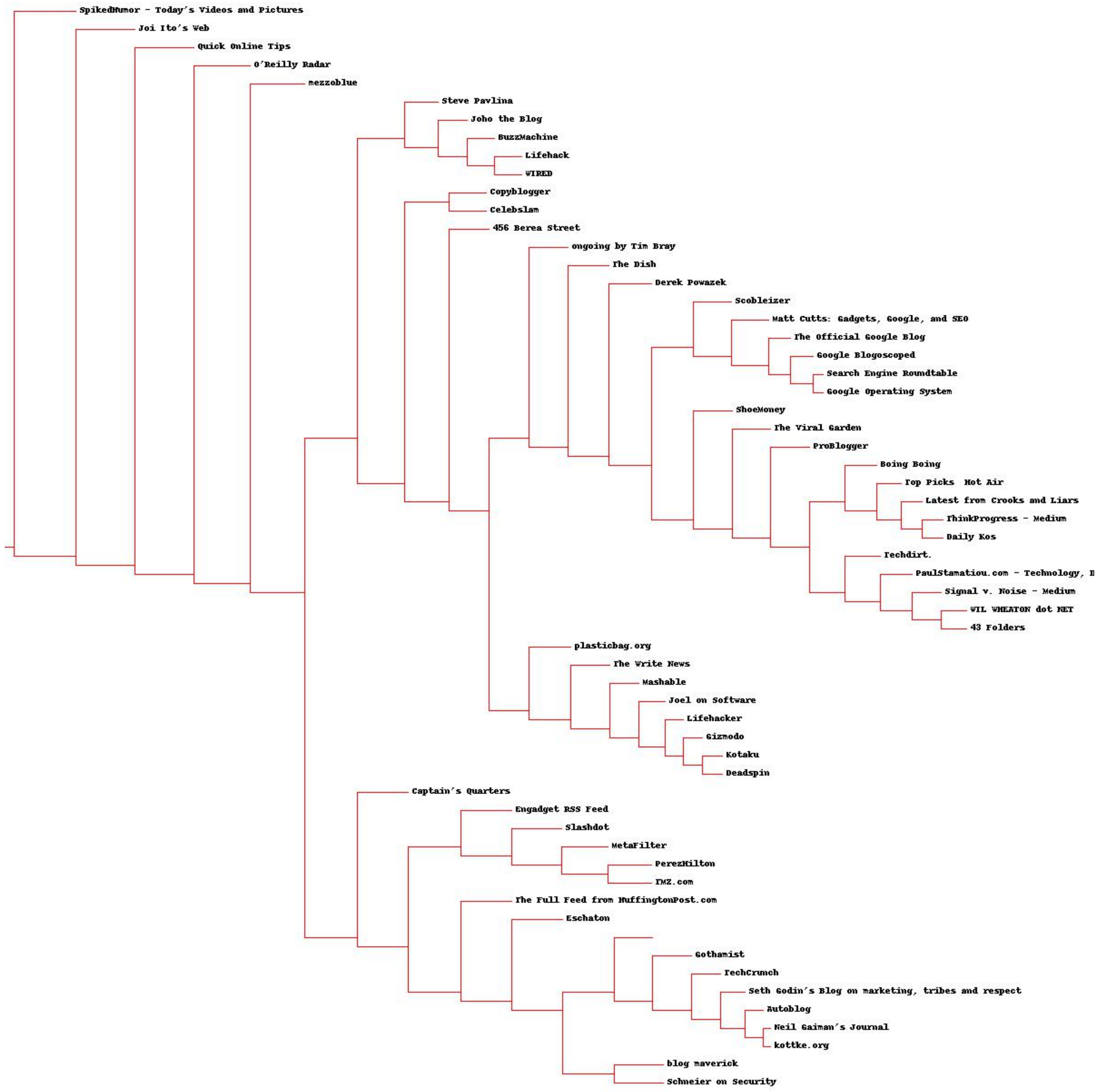
	"china"	"kids"	"music"	"yahoo"
Gothamist	0	3	3	0
GigaOM	6	0	0	2
Quick Online Tips	0	2	2	22

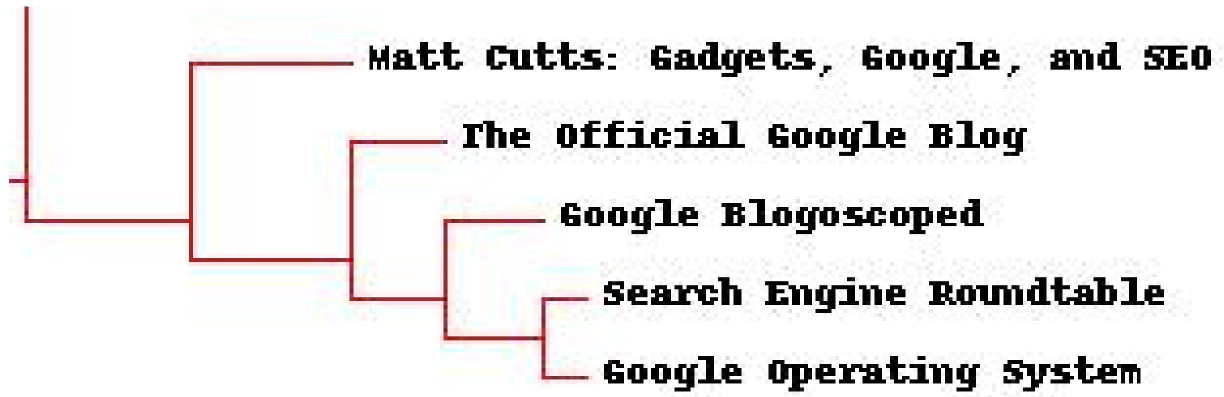
```
import clusters

blognames, words, data = clusters.readfile('blogdata.txt')
clust=clusters.hcluster(data)
```

Visualizing Clusters - Dendograms




SpikedHumor - Today's Videos and Pictures
Joi Ito's Web
Quick Online Tips
O'Reilly Radar
mezzoblue
Steve Pavlina
Joho the Blog
BuzzMachine
Lifehack
WIRED
Copyblogger
Celebslam
456 Berea Street
ongoing by Tim Bray
The Dish
Derek Powazek
Scobleizer
Matt Cutts: Gadgets, Google, and SEO
The Official Google Blog
Google Blogoscoped
Search Engine Roundtable
Google Operating System
ShoeMoney
The Viral Garden
ProBlogger
Boing Boing
Top Picks Hot Air
Latest from Crooks and Liars
ThinkProgress - Medium
Daily Kos
Techdirt.
PaulStamatiou.com - Technology, D
Signal v. Noise - Medium
WIL WHEATON dot NET
43 Folders
plasticbag.org
The Write News
Mashable
Joel on Software
Lifehacker
Gizmodo
Kotaku
Deadspin
Captain's Quarters
Engadget RSS Feed
Slashdot
MetaFilter
PerezHilton
TMZ.com
The Full Feed from HuffingtonPost.com
Eschaton
Gothamist
TechCrunch
Seth Godin's Blog on marketing, tribes and respect
Autoblog
Neil Gaiman's Journal
kottke.org
blog maverick
Schneier on Security


Matt Cutts: Gadgets, Google, and SEO
The Official Google Blog
Google Blogoscoped
Search Engine Roundtable
Google Operating System

Cluster Object

```
class Bicluster:
  def __init__(self,vec,left=None,right=None,distance=0,id=None):
    self.left = left
    self.right = right
    self.vec = vec
    self.id = id
    self.distance = distance
```

Clustering - I

```
def hcluster(rows, distance=pearson):
  distances = {}
  currentclustid = -1
  # Clusters are initially just the rows
  clust = [Bicluster(rows[i], id = i) for i in range(len(rows))]

  while len(clust) > 1:
    lowestpair = (0, 1)
    closest = distance(clust[0].vec, clust[1].vec)
    # loop through every pair looking for the smallest distance
    for i in range(len(clust)):
      for j in range(i+1, len(clust)):
        # distances is the cache of distance calculations
        if (clust[i].id, clust[j].id) not in distances:
          distances[(clust[i].id, clust[j].id)]=
                                             distance(clust[i].vec, clust[j].vec)
        d=distances[(clust[i].id, clust[j].id)]

        if d < closest:
          closest = d
          lowestpair = (i, j)
```

End loop

End loop

Clustering - II

```
    # calculate the average of the two clusters
    vec1 = clust[lowestpair[0]].vec
    vec2 = clust[lowestpair[1]].vec
    mergevec=[(vec1[i]+vec2[i])/2 for i in range(len(vec1))]

    # create the new cluster
    newcluster = Bicluster(mergevec,
                           left=clust[lowestpair[0]],
                           right=clust[lowestpair[1]],
                           distance=closest,
                           id=currentclustid)

    # cluster ids that weren't in the original set are negative
    currentclustid-=1
    del clust[lowestpair[1]]
    del clust[lowestpair[0]]
    clust.append(newcluster)

return clust[0]
```

End while loop