```
         for(int i = 0;i<n;i++) {
            int xpix = panel.xToPix(x[i]);
            int ypix = panel.yToPix(y[i]);
            for(int s = 0;s<m;s++) {
               int j = linkFrom[i*m+s];
               int xpixj = panel.xToPix(x[j]);
               int ypixj = panel.yToPix(y[j]);
               // draw link
               g.drawLine(xpix, ypix, xpixj, ypixj);
            }
         }
         g.setColor(Color.red);
         for(int i = 0;i<n;i++) {
            int xpix = panel.xToPix(x[i])-pxRadius;
            int ypix = panel.yToPix(y[i])-pyRadius;
            // draw node
            g.fillOval(xpix, ypix, 2*pxRadius, 2*pyRadius);
         }
      }
   }
}
```

Problem 14.16 Preferential attachment model

(a) Write a target class that uses the `PreferentialAttachment` class and continuously creates new networks until stopped by the user (so we can compute averages over many networks). To speed up the computation, make it possible to optionally display the networks. The program should output the average degree distribution $D(\ell)$.

(b) Estimate the exponent α defined by $D(\ell) \sim \ell^{-\alpha}$ for $N = 100$ and $m = 2$. Repeat for $N = 500$. Does the exponent α change? If time permits, consider $N = 10{,}000$. Does α depend on m?

(c) Modify `PreferentialAttachment` so that the ℓ links are made randomly so that the number of links a node already has is irrelevant to adding a link. What functional form does the link distribution have now? Is this model equivalent to the Erdös–Rényi model?

*(d) Write a method to compute the clustering coefficient, which is defined in Figure 14.3. Plot $\ln C(N)$ versus $\ln N$ for both the preference attachment model and the Erdös–Rényi model. Compare and discuss your results in terms of the visual appearance of the networks. ■

Problem 14.17 Watts–Strogatz network

(a) Write a class to create a Watts–Strogatz network. Begin with $N = 100$ nodes which you can visualize as equally spaced on a circle. (Their actual position is irrelevant.) Place links between the $2m$ nearest neighbors. Thus, if $m = 1$, then only the nearest neighbors are linked. If $m = 2$, then the nearest and next nearest neighbors are linked. Next, write a method to go through each link and then with probability p, break the link connection at one end and reconnect it to another node at random.

(b) Compute the degree distribution as a function of m for several values of p. Discuss your results.

(c) As we increase p, the networks become more and more random. There is a transition from a network where the path length $\ell \sim N$ to one where $\ell \sim \ln N$. This transition occurs when $Np^{1/d} \sim 1$, where d is the dimension of the original lattice before rewiring ($d = 1$ for a circle). Draw a number of networks with different values of N and p and use this visualization to explain the dependence of ℓ on N.

*(d) Write a method to compute the clustering coefficient C. Plot C versus $\ln p$ for $N = 100$ and $m = 2$. Repeat for larger N. ■

Problem 14.18 A model of a social network

In many social situations, we notice groups of people who interact closely with each other, but not necessarily with other groups. Usually, those in a group have some common interest or personal attribute. How can we model this situation? People do not usually become friends with other people just because they have many friends already (the preferential attachment mechanism). Instead, they choose someone to interact with and a friendship is established with some probability. A simple model is given by the following rule. As each node is added to a system, choose m existing nodes at random, and with probability p establish a link. This process will create a number of clusters of linked nodes. We can imagine that there is a possibility of a phase transition between the existence of a giant cluster that contains a large fraction of the nodes and a situation where all the clusters are small. This model was analyzed by Zalányi et al.

(a) Write a class to model this random attachment model and compute the degree distribution as well as the cluster distribution. Consider at least $N = 1000$ nodes and measure $D(\ell)$, the degree distribution, for $m = 2, 3$, and 5 and $p = 0.1$ and $p = 0.9$. Average over at least ten trials. You should not find power law behavior for $D(\ell)$. Explain why this behavior is expected.

(b) Compute $D(\ell, t)$, the number of links connected to a node as a function of t, the time when the node is added. We would expect nodes added in the beginning to have more links than those at the end. Describe and discuss the functional form of $D(\ell, t)$.

(c) Consider $m = 5$ and generate many networks for different values of p. Determine the cluster distribution. A giant cluster exists when the largest cluster is at least three times larger than the next largest cluster. Estimate the value of p for which the giant cluster first appears. You should find an approximate power law cluster distribution only at the transition. What is the exponent of the power law?

(d) How does the value of p at the transition change with m? Explain your results.

(e) Consider $m = 1$ and generate networks for many values of p. Determine the cluster distribution. You should find an approximate power law distribution for all values of p. What are the exponents for the power law? Why do you think there is not a phase transition for $m = 1$? Consider the possibility of two clusters merging for different values of m. ■

14.5 ■ GENETIC ALGORITHMS

Many people find it difficult to accept that evolution is sufficiently powerful to generate the biological complexity observed in nature. Part of this difficulty arises from the inability