Figure 21.1  (a) A graph with four connected components: \{a, b, c, d\}, \{e, f, g\}, \{h, i\}, and \{j\}.  
(b) The collection of disjoint sets after each edge is processed.
**Connected-Components**$(G)$

1. for each vertex $v \in V[G]$
2. do **Make-Set**$(v)$
3. for each edge $(u, v) \in E[G]$
4. do if **Find-Set**$(u) \neq \text{Find-Set}(v)$
5. then **Union**$(u, v)$

**Same-Component**$(u, v)$

1. if **Find-Set**$(u) = \text{Find-Set}(v)$
2. then return **TRUE**
3. else return **FALSE**
Figure 21.2  (a) Linked-list representations of two sets. One contains objects $b$, $c$, $e$, and $h$, with $c$ as the representative, and the other contains objects $d$, $f$, and $g$, with $f$ as the representative. Each object on the list contains a set member, a pointer to the next object on the list, and a pointer back to the first object on the list, which is the representative. Each list has pointers head and tail to the first and last objects, respectively. (b) The result of $\text{UNION}(e, g)$. The representative of the resulting set is $f$.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Number of objects updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{MAKE-SET}(x_1)$</td>
<td>1</td>
</tr>
<tr>
<td>$\text{MAKE-SET}(x_2)$</td>
<td>1</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$\text{MAKE-SET}(x_n)$</td>
<td>1</td>
</tr>
<tr>
<td>$\text{UNION}(x_1, x_2)$</td>
<td>1</td>
</tr>
<tr>
<td>$\text{UNION}(x_2, x_3)$</td>
<td>2</td>
</tr>
<tr>
<td>$\text{UNION}(x_3, x_4)$</td>
<td>3</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$\text{UNION}(x_{n-1}, x_n)$</td>
<td>$n - 1$</td>
</tr>
</tbody>
</table>

Figure 21.3  A sequence of $2n - 1$ operations on $n$ objects that takes $\Theta(n^2)$ time, or $\Theta(n)$ time per operation on average, using the linked-list set representation and the simple implementation of $\text{UNION}$. 
Figure 21.4 A disjoint-set forest. (a) Two trees representing the two sets of Figure 21.2. The tree on the left represents the set \{b, c, e, h\}, with c as the representative, and the tree on the right represents the set \{d, f, g\}, with f as the representative. (b) The result of \textsc{Union}(e, g).

Figure 21.5 Path compression during the operation \textsc{Find-Set}. Arrows and self-loops at roots are omitted. (a) A tree representing a set prior to executing \textsc{Find-Set}(a). Triangles represent subtrees whose roots are the nodes shown. Each node has a pointer to its parent. (b) The same set after executing \textsc{Find-Set}(a). Each node on the find path now points directly to the root.
**MAKE-SET**($x$)

1. $p[x] \leftarrow x$
2. $\text{rank}[x] \leftarrow 0$

**UNION**($x$, $y$)

1. $\text{LINK} (\text{FIND-SET}(x), \text{FIND-SET}(y))$
\[ \textbf{LINK}(x, y) \]
1. \textbf{if} \( \text{rank}[x] > \text{rank}[y] \)
2. \quad \textbf{then} \( p[y] \leftarrow x \)
3. \quad \textbf{else} \( p[x] \leftarrow y \)
4. \quad \textbf{if} \( \text{rank}[x] = \text{rank}[y] \)
5. \quad \textbf{then} \( \text{rank}[y] \leftarrow \text{rank}[y] + 1 \)

\[ \textbf{FIND-SET}(x) \]
1. \textbf{if} \( x \neq p[x] \)
2. \quad \textbf{then} \( p[x] \leftarrow \textbf{FIND-SET}(p[x]) \)
3. \quad \textbf{return} \( p[x] \)