Disjoint Sets
CS 320, Fall 2017

Dr. Geri Georg, Instructor
georg@colostate.edu
Data Structure

Collection of disjoint, dynamic, sets
• Each set has a representative
• Representative is a member of the set
• Can change if set is modified
• Complexity:
  • \( n \), the number of Make-Set operations
  • \( m \), the total number of Make-Set, Union, and Find-Set operations
Operations

**MAKE-SET(x) → S_x = \{x\}**
- x is representative and not in another set

**UNION(x, y) → S_z = S_x \cup S_y**
- representative is any element in either set, original sets “destroyed”

**FIND-SET(x) → S_x**
- x ∈ S_x

Complexity:
- n objects, & num Make-Set operations
- m, total num Make-Set, Union, and Find-Set operations
Connected Components

CONNECTED-COMPONENTS (G)
for each vertex v ∈ G.V
MAKE-SET(v)
for each edge (u,v) ∈ G.E
if FIND-SET(u) ≠ FIND-SET(v)
  UNION(u,v)
SAME-COMPONENT (u,v)
  if FIND-SET(u) == FIND-SET(v)
    return TRUE
  else return FALSE

G is an undirected graph
Linked Lists

Complexity of Make-Set? Find-Set? Union?
In lieu of recitations – 320 Office Hours

<table>
<thead>
<tr>
<th></th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thurs</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>Cole</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Dr. Georg</td>
<td></td>
<td>Dr. Georg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>Dr. Georg/ Jim</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Ali</td>
<td></td>
<td>Jim</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Cole</td>
<td></td>
<td>Ali</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Cole/ Shannon</td>
<td></td>
<td></td>
<td>Ali</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Shannon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Dr. Georg</td>
<td></td>
<td>Dr. Georg</td>
<td></td>
<td>Jim</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Upcoming -- Check Progress page, Piazza postings for updates**
• MST Worksheet due beginning of class Wed
Micro-Survey – 1

When to use Fibonacci heaps?

Good when:

• number of Extract-Min and Delete small compared to other ops
• dense graphs: lots of Decrease-Key ops

Meaning of disjoint:

Elements in one set cannot also be in another set. Ex: the set of trees, 1 per edge, we create when we begin Kruskal’s algorithm to find an MST – initially each edge is only in 1 set, and as we merge edges into sets, each edge is still only in 1 set.
Micro-Survey – 2

Disjoint set complexity:

Assume n elements, n Make-Set calls, and n-1 Union calls, and a linked list implementation:

**Make-Set:** $\Theta(n)$

$m = n + n-1 = 2n - 1$

\[
\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = \Theta(n^2)
\]

Overall we have $\Theta(n) + \Theta(n^2) = \Theta(n^2)$

BUT, $m = 2n-1$ so we can say that $\Theta(n^2)$ is over the m ops and divide $n^2/2n-1$ using big-O arithmetic and arrive at an amortized complexity of $\Theta(n)$.
Amortized Time

We have $\Theta(n^2)$ time, and $m = 2n - 1$ so we can talk about amortized time:

$$\frac{n^2}{2n - 1},$$
so each op on average is $\Theta(n)$
Trees instead of Lists

Union: Append the smaller lower size tree as a child of the larger.

Apply path-compression to re-direct child nodes to the root.
Image Credits

mazefromdisjtssets: http://web.eecs.utk.edu/~plank/plank/classes/cs302/Notes/Disjoint/
