CSE 373 Optional Section
Disjoint Sets & Homework 4

February 13, 2014
Nicholas Shahan & A. Conrad Nied
Agenda

• Disjoint Sets Review
• Homework 4 Examples
Disjoint Sets Review

OUTBREAK trailer

Interactive Example
HW4: Representation?

• What data structures do we want?
• Java Collections
• HashMap, HashSet, Arrays, ArrayList, etc.
  – Examples:
    • Two arrays, for vertical and horizontal walls
    • A Class for maze walls, stored in a HashSet
HW4: What Numbers Matter?

• How many rooms are in the maze?
• Height or Number of Rows
• Width or Number of Columns
• How many interior walls?
• How many exterior walls?
HW4: By The Numbers

• For a maze with 4 rows and 5 columns:
  • 20 total rooms
  • 49 walls
  • 31 interior walls
    – 16 vertical interior walls (4 x 4)
    – 15 horizontal interior walls (3 x 5)
HW4: By The Numbers

• For a maze with 3 rows and 7 columns:
  • 21 total rooms
  • 52 walls
  • 32 interior walls
    – 18 vertical interior walls (3 x 6)
    – 14 horizontal interior walls (2 x 7)
HW4: Consistent Identification

One Example: Each room is responsible for knowing its set and its walls left of and below it.
HW4: Example Maze

height = 3, width = 4
HW4: Example Maze

Number the rooms
HW4: Example Maze

After uniting all of the nodes, the Disjoint Sets array looks like this:
HW4: Example Maze

The walls to the left of each room are valued as such (1 means that the wall is in the maze)
HW4: Example Maze

Likewise, for the horizontal walls below each room:
HW4: Example Maze

All together, the numbers for the rooms and the walls look like this:
HW4: Don’t Create Cycles

Every time you remove an interior wall it is possible that you created a cycle:
HW4: Example Maze

With 3 arrays that look like this:

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set</td>
<td>-1</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>9</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>wall_h</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>wall_v</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
HW4: MazeBuilder

• Find wall to remove
  – Find at random, rooms, walls, and/or neighbors
  – Make sure you want to remove it
    • Its neighbors are disjoint but adjacent
    • Its not exterior

• Check if you should continue removing walls

“Until all are one...”
HW4: MazeBuilder

To print your mazes:
• Make the top row of walls
• For each row
  – Make a row of rooms and vertical walls
  – Make a row of horizontal walls below this row of rooms
• Make the bottom row of walls
• Make sure that you have an opening at the entrance and exit
HW4: Locations

Checking for adjacency where $w$ is the width and $i$ is the index of your room. Make sure to not modify or jump over exterior walls.

\[
\begin{align*}
4i \equiv 1 & \pmod{5} \\
i & \equiv 6i + 1 \\
1 & \equiv 6i + w
\end{align*}
\]
HW4: MazeSolver

The solution is 0 4 8 9 5 6 10 11