COMP261 Lecture 3

Graphs 2 of 3

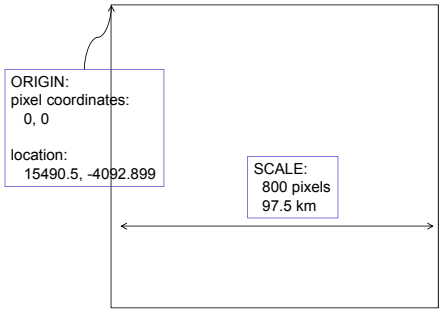
Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui CAPITAL CITY UNIVERSITY

Locations (The Hardest Part of A1?)

Three representations of a location/place/point

- latitude/longitude
  - what you need for locations on a sphere
  - This is what is in the data
- Location: x/y coordinates in kilometers
  - assume Auckland region is a flat plane (not quite true, but good enough)
  - This is what you need for finding shortest paths
- Point (x/y in pixels):positions on the screen in pixels
  - for drawing lines
  - mouse click positions
- → need to translate
  - lat/lon → x/y kms (use an appropriate formula)
  - x/y kms ↔ pixel coords
    - depends on current origin and scale!

Location vs pixels.



Look at the Location class!

Data Structures for Graphs

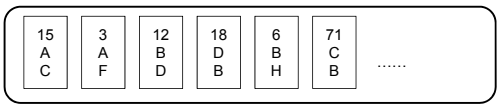
Need to represent the nodes and the edges

- Mathematical definition:
  set of nodes V = $\{v_i\}$,
  set (or bag) of pairs (ordered or unordered) of nodes E = $\{(v_i, v_j)\}$
- Possible Data Structure:
  - Graph = set of Node objects Set<Node> nodes;
    & set/bag of Edge objects Set<Edge> edges;
  - Node = info about the node (label, etc)
  - Edge = info about the edge and its two nodes
    either two fields: node1, node2, or
    or Set<Node> (just 2 elements)
    - Issue: when are two edges counted as equal?

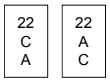
Possible Data Structure

- Nodes

- Edges



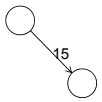
- Is this a good data structure?
  - How could we tell?
  - Depends on what we want to do with the graph.



More Data Structures for Graphs

What do we need to do with a graph?

- Common actions on a graph:
  - list all the nodes
  - list all the edges
  - find all the neighbours of a node
    (nodes at other end of edge out of this node)
  - find all the nodes of which this node is a neighbour
    (nodes that have an edge to this node)
  - determine whether two nodes are connected or not
  - find the label/weight on the edge between two nodes



maybe same if undirected

What do we need for the Road Map?

Auckland Roads:

- Display the map (and zoom in/out)
  - ➢ *must access all the road elements and intersections*
- Select roads (by name)
  - – Show all roads matching what is typed so far
  - – Highlight road on map.
  - ➢ *must access roads by name, (and by prefix of name!)*
  - ➢ *must access description of all the road elements.*
- Select intersections (by mouse click)
  - – Highlight intersection
  - ➢ *must access intersection by its location (nearest match)*
  - – Display names of roads at intersection.
  - ➢ *must access all the roads for a given intersection*

Reading data more efficiently

- File
- FileReader
- BufferedReader

```
// Read file line by line
File roadFile = new File(dataDirectory+"roadID-roadInfo.tab");
BufferedReader data = new BufferedReader(new FileReader(roadFile));
String line = data.readLine();

// Process each line using split method
String[] values = line.split("\t");
 int n = Integer.parseInt(values[0]);
 double d = Double.parseDouble(values[1]);
```

More Data Structures for Graphs

- Structures for explicit graphs:
  - "Traditional" Adjacency matrix
  - "Traditional" Adjacency lists
  - Object and Collection based representations
- Structures for implicit graphs:
  - Node objects and neighbour functions.