CS2040 Lab 10
MST

One Day Assignment 8 – Islands

- Easier way of looping through all 4 directions (instead of hardcode):
- `int[][] move = {{-1, 0}, {0, 1}, {1, 0}, {0, -1}};`
- `for (int i = 0; i < 4; i++) {`
  - `int nextRow = row + move[i][0];`
  - `int nextCol = col + move[i][1];`
  - `// check for out of bounds etc. here`
- `}`
- The above code traverses up, right, down and left, in that order

Lab 10 – MST

- Two different forms of MST algorithms are covered: Prim’s and Kruskal’s
    - Prim’s tends to be used alongside an Adjacency List (or an Adjacency Matrix in the cases of near complete graphs), while Kruskal’s tends to be used alongside an Edge List
    - Prim’s uses a priority queue as well, while Kruskal’s uses a UFDS
- Examples provided in lectures

Take Home Assignment 4 – Millionaire Madness

- Uses a 2D grid (code example for moving up/down etc. may help)
- Need to reach the lower right corner of the grid from the upper left corner
- Each cell has a specific height
- A ladder is needed when going up in height (the length of the ladder must be >= the difference in height)
- A ladder is not required when going down in height
- Find the minimum ladder length needed

Take Home Assignment 4 – Millionaire Madness

- Route for last sample input (red -> blue -> green, endpoints in bold):
-

10	11	12	13	**14**
11	20	16	17	**16**
12	10	18	**21**	**24**
14	10	14	14	22
16	18	20	20	25
25	24	22	10	25
26	27	28	21	**25**

- Can be solved via correct graph modelling and a *specific* MST algorithm

Take Home Assignment 4 – Millionaire Madness

Take Home Assignment 4 – Dominos

- Knocking down one domino manually may result in a few other dominos being knocked down as well
- These dominos may in turn cause other dominos to knock over as well
- Find the minimum number of dominos that must be knocked down manually so that all dominos are knocked down
- Trying to visualise the solution on a general graph might be difficult; can consider converting the graph to a specific type of graph that’s simpler to visualise with

One Day Assignment 9 – Lost Map

- Given the shortest path (in terms of distance) between any two villages on a map, find all the roads that make up the original set of roads
  - Graph is a complete graph, and hence consists of a lot of edges
- The following information is given (explicitly, or deduced from the problem description):
  - The original set of roads form a connected, weighted tree
  - Distance of any road (u, v) in the map is > 0, unless (u == v)
- Note: using Scanner can still work here, but it is recommended to use buffered IO for this problem (saves 3+ seconds of CPU time, out of 8)