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
- Alternatively, can use two arrays: int[] dx and int[] dy

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
- There are other interesting MST algorithms, e.g. Boruvka and Reverse-Delete

Take Home Assignment 4 – Millionaire Madness

- Also uses a 2D grid
- 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

* There are also other solutions but may not be that easy to figure out.

Take Home Assignment 4 – Millionaire Madness

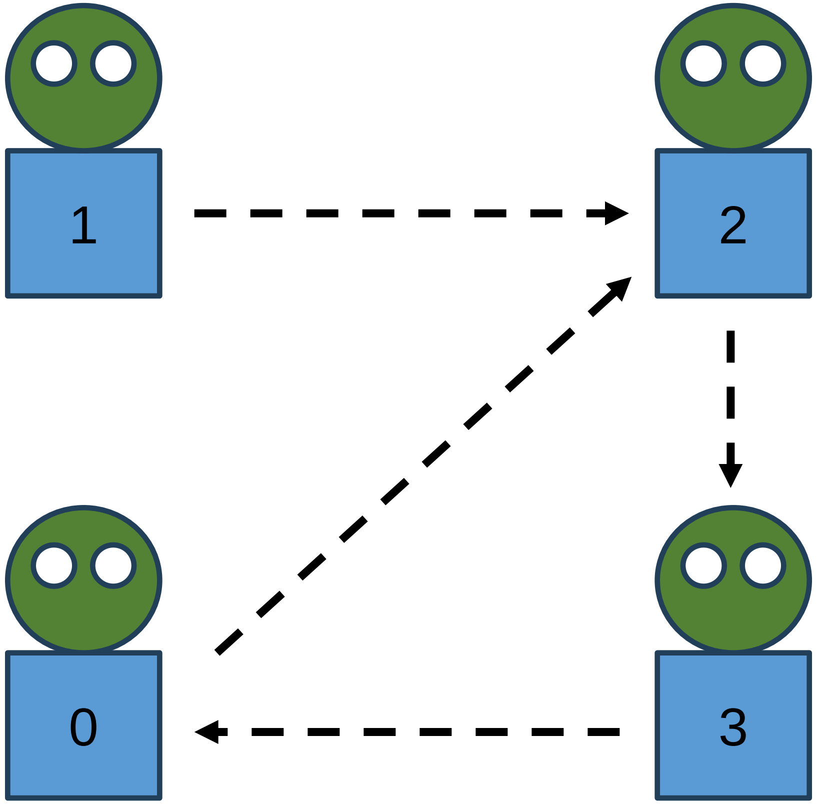
Take Home Assignment 4 – Bots

- We can issue the program to a bot which will send the message to other bots
- These bots that received the message may in turn send the message to other bots
- Find the number of bots that the program must be issued to so that all bots can receive the message to go to sleep mode
    - Min no. of solobots
    - Min no. of botnets

Take Home Assignment 4 – Bots

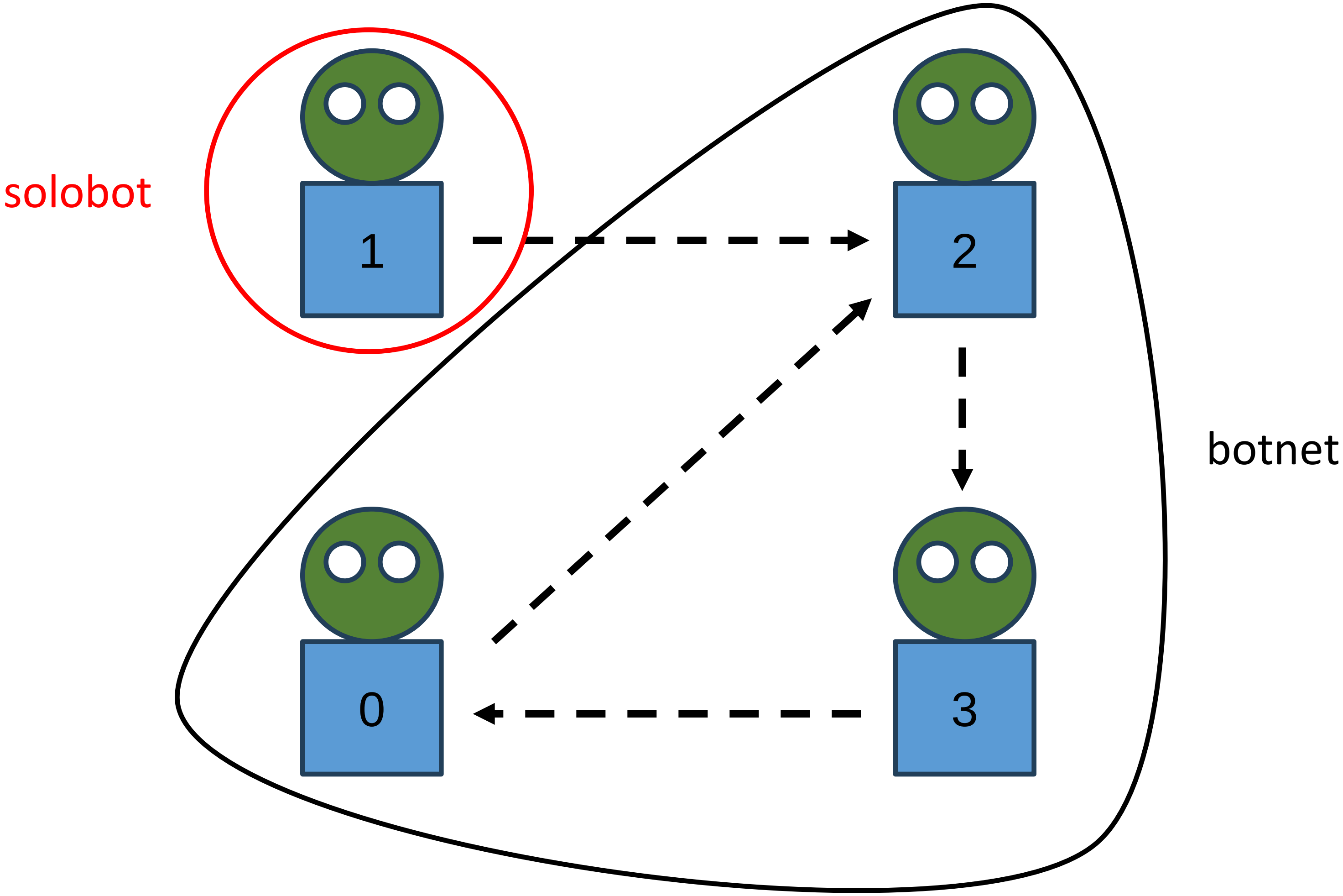
- If **A** can signal **B**, does it mean that **B** can signal **A**?
    - See sample input 1 and 2
- If **A** sends a message to **B**
    - **A** signals **B** directly
    - **A** signals **B** through a sequence of bots, e.g. **A** signals **C, C** signals **B**
- Any 2 bots in the same botnet can message each other
    - If there is a bot **O** outside the botnet, either cannot send a message to **O** or from **O** to botnet i.e one direction either in or out of botnet, if at all
- Solobot is its "own botnet"
    - No other bot to message both ways with
- How should you represent the graph?
    - What about the vertices/ edges? What do you need to know about your graph?

Take Home Assignment 4 – Bots



Which bot(s) should you issue the program to?

Take Home Assignment 4 – Bots



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
- 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)