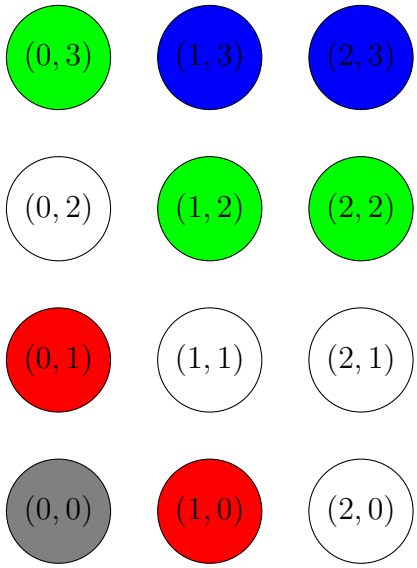
Parallel Processing – task 3

19.11.2015

Design and implement the synchronization protocol for the following problem. Use the MPI library.

Problem

1. We have $n \cdot m$ cities, arranged in an $n \times m$ grid (see picture below).



2. They are fighting a terminal disease, which is spreading in the region.
3. Each city can be in one of the following states: Neutral, Protected, Producing, Contagious or Dead.
4. If a city is Neutral, it does nothing (and waits for something to happen).

5. If a city is PRODUCING, it means that the scientists are currently developing the vaccination. It takes some random amount of time (choose the appropriate value). Then the city sends transports to all neighboring cities (in the directions: N, S, E, W) at the same time (of course boundary cities have less neighbors). After sending transports, the city becomes PROTECTED.

6. If the transport of vaccination reaches a city, which is NEUTRAL, it becomes PRODUCING. The other types of cities do not change their state after obtaining a transport.

7. If a city is CONTAGIOUS, it waits some random amount of time (choose the appropriate value) and then spreads the disease to all neighboring cities. After that, the city becomes DEAD.

8. If a disease spreads in a NEUTRAL city, it becomes CONTAGIOUS. No states are changing in the remaining cases.

9. This is repeated until all cities become either DEAD or PROTECTED.

10. Finally, print the total number and names (process id's) of cities that are DEAD and PROTECTED in a manner similar to the following one (in any order):
```
DEAD: 3 (0,0) (0,1) (1,1)
PROTECTED: 3 (1,0) (2,1) (2,0).
```

Technical aspects

1. Each city is represented by a single process. The name of the city is this process's id.

2. If a city becomes PROTECTED or DEAD, we can terminate the process.

3. Use a virtual topology.

4. Your program should take a file as an input argument. This file's format is:

[n]
[m]

[Coordinates of PRODUCING cities.
Each city is given in a separate line,
x- and y-coordinates are separated by a comma.]

[Coordinates of CONTAGIOUS cities
Each city is given in a separate line,
x- and y-coordinates are separated by a comma.]

A sample input file is given in `disease.input`.

5. Initialize the states of the cities according to the input file. All the remaining cities are initially NEUTRAL.

6. You should analyze the messages in the same order as they arrive. You should not give any priority to a specific neighbor or type of message.

7. Include the comments describing the synchronization process.

8. Print the current status of the processes in the console.

9. Your submission should contain a file with the code and a Makefile. Typing `make compile` should compile your application, while typing `make run [file]` should call your program with the parameter with [file] being the input file name (assume that the name and file format are correct).

Deadline for submitting your solution: 01.12.2015 (23:59)