Algorithms and Data Structures 1

Student number: s215158
Hand-in: Exam simulation

1 Exercise

1.1

In order to model the zombie outbreak as a graph we can simply imagine that every person is a node and the edges, represented by the infection (p_1, p_2), are represented as directed edges from node p_1 to node p_2.

1.2

Supposing that the graph uses adjacency list we can find all the nodes with in-degree zero simply searching in the adjacency list representation of the reverse graph to find all patient zeros. This method is also used in the algorithm to find a topological sorting for a directed graph, if it exists. The quickness of the algorithm depends on the number of people P that are stored in the list. To find all patients zero we will need to fully traverse the list so we will have a running time of: $O(P)$.

1.3

Given a zombie outbreak, we can determine if the zombie outbreak is inconsistent or consistent running a topological sort on the graph. The outbreak is consistent if the graph is a DAG otherwise it is inconsistent. This method will work since the definition of "inconsistent" from the exercise reflects the one of having a cycle and therefore not being able to do a topological sort on the graph. The running time is: $O(P + I)$ if we use the second algorithm presented in the lecture.

1.4

After we have updated our graph with edge weights we can run the shortest path algorithms for DAGs. This will give us the correct shortest path because, by definition, when a new edge u is added to the the tree T the distance to the new node u will be the previous distance plus the edge e to u. Assuming that we relax all the edges every time we add a new vertex then the distance to u will always be the shortest. The running time is: $O(P + I)$ if we follow the implementation given in the lecture.