Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Institut für Theoretische Informatik
Peter Widmayer
Thomas Tschager
Antonis Thomas

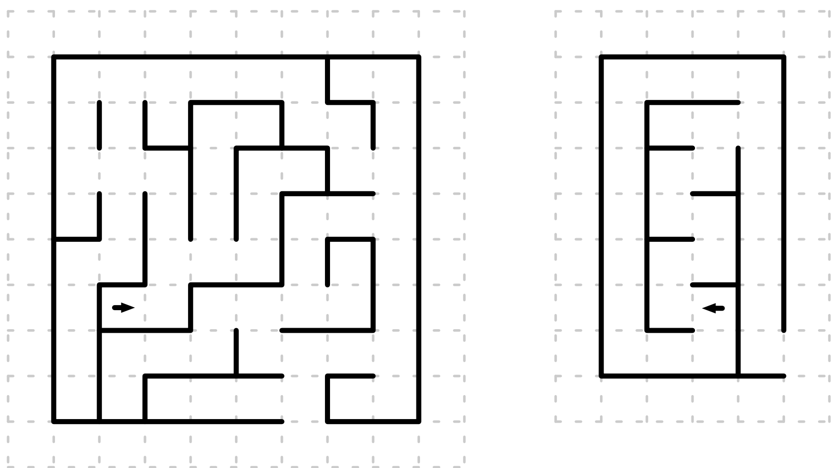
4th May 2016

Datenstrukturen & Algorithmen Exercise Sheet 10 FS 16

Exercise 10.1 *Path Planning in Labyrinths.*

You are given a labyrinth as a drawing on squared paper, like in the example below. At the marked point there is a robot facing the direction indicated by the arrow. The question is how fast the robot can escape the labyrinth. The robot can travel "forward" one square in the direction it is facing within 3 seconds. To stop after a forward movement takes 2 seconds. While standing still, the robot can rotate 90 degrees, which costs 2 seconds. The robot does not need to stand still between two consecutive forward movements (although it could, but that would take more time).

In the examples below, the robot needs 113s and 79s to escape.



a) Model the above problem as a shortest path problem. Describe how to represent the labyrinth as a graph such that the length of the shortest path in the graph equals the time that the robot needs to escape.

b) What is an efficient algorithm to solve this problem?

c) Which running time in dependency of the number of squares of the labyrinth does this algorithm have when it is applied to the graph constructed in a)?

Exercise 10.2 *Variants of Shortest Path Problems.*

Let $G = (V, E, w)$ be a directed, weighted graph with positive edge weights and $s, t \in V$ be two vertices. Design an efficient algorithm for the following variants of the shortest path problem. Also, analyze the running time.

a) Let $k \in \mathbb{N}$ be arbitrary. Among all possible paths from s to t with at most k inner vertices (i.e., with at most $k+1$ edges) we are looking for the one with minimum length.

b) Beside the shortest path from s to t we are also searching for the second shortest path.

c) We are searching for the number of all different shortest paths from s to t.

Exercise 10.3 *Closeness centrality (Programming exercise).*

In this exercise we are going to compute the *closeness centrality* measure for undirected graphs. This measures how close a vertex is to all other vertices in the graph. For the purpose of this exercise, we define closeness centrality for a vertex v as

$$C(v) = \sum_{u \in V} d(v, u)$$

where V is the set of vertices of the graph and $d(v, u)$ is the length of the shortest path connecting v and u. If there exists no path between two vertices v and w, we define $d(v, w) = 0$. You can assume that the input graph is connected, minus a few isolated nodes. Thus, an isolated vertex v has $C(v) = 0$. The smaller $C(v)$ is (ignoring the isolated vertices), the more central node v is.

(Note: our definition of the measure is the inverse of the most common definition found in literature but we stick to it for the purpose of the exercise)

Input The input in this exercise is based on real-life data. It is based on the collaborator graph of renowned mathematician Paul Erdős. The input contains the description of a graph with 511 nodes. Every node represents a coauthor of Erdős. Every edge between two nodes represents that the respective authors have published a paper together. Information on the file can be found here: `http://wwwp.oakland.edu/enp/thedata/`. The actual file that we use as input is derived from this: `http://files.oakland.edu/users/grossman/enp/erdos1graph.html`. The graph has been slightly manipulated for the purpose of this exercise.

Output For each node v (in lexicographic order, as in the input file) of the graph output $C(v)$.

Directions We provide you with a template. It contains the necessary code to read the input. The public testset is the same as the private testset for this exercise.

We expect you to implement the Floyd-War shall algorithm so that you can compute the closeness centrality.

Example We give the names of the 5 researchers with the smallest centrality, sorted incrementally. The names appear as given in the input file.

Top 5:

```
GRAHAM, RONALD LEWIS
ALON, NOGA M.
FUREDI, ZOLTAN
SOS, VERA TURAN
BOLLOBAS, BELA
```

You should not be surprised to know that all of these names belong to famous mathematicians.

Hand-in: Wednesday, 11th May 2016 in your exercise group.