Guide to the coursework

Dr. Klaus Draeger

March 14, 2021

Networks

- The coursework deals with **flows** on **networks**
- A network is a directed graph . . .
  - Nodes
  - Edges $m \to n$ for some pairs of nodes m, n
- With additional data:
  - Two special nodes, the **source** and the **target**
  - Each edge has a **capacity** (how much flow it can accommodate)

Flows

- The coursework deals with **flows** on **networks**
- A **flow** associates a number with each edge
  - At least 0, at most the edge's capacity
  - Flow along an edge $m \to n$ flows **out of** m and **into** n
  - For each node n except the source and target, the total flow **into** n must equal the total flow **out of** n
  - No flow into the source or out of the target
- We want a flow which **maximises** the flow out of the source.

What to do: part 1

- Choose a suitable **data structure** to represent networks
  - The lecture introduced some ways of representing **graphs**
  - Think about how these can be adapted to accommodate additional data
    (capacity and flow for each edge)
- Implement your data structure
- You will also need a method for printing a graph
  (for debugging and showing steps of your algorithm)

What to do: part 2

- Implement a parser to read a network from a file
- The file format is:
  - One line containing the number N of nodes
    - Nodes are numbered $0, \ldots, N-1$
    - The source is always node 0
    - The target is always node $N-1$
  - Several lines describing edges: for example,
    2 4 5
    means that there is an edge
    - from node 2
    - to node 4
    - with capacity 5

What to do: part 3

- Choose an algorithm to compute the maximum flow
- We will describe one (Ford-Fulkerson) in the lecture
- Implement your algorithm
- Test it using several examples, such as the benchmarks provided
- Write your report:
  - A short explanation of your choice of data structure and algorithm
  - A run of your algorithm on a small benchmark example
  - Timing analysis of your implementation