# Advanced Algorithms and Datastructures

## Exam Notes

*Author:*
Jenny-Margrethe Vej – 250986 – rwj935

Block 4, April-June, 2014

# Contents

# 1 Maximum Flow

Wuuuh, Max Flow!! You Rock! Kick some ass!

### 1.0.1 Flow Networks and Flow

Let $G = (V, E)$ be a flow network with a capacity function $c$. Let $s$ be the source of the network, and let $t$ be the sink. A flow in $G$ is a real-valued function $f : V \times V \to \mathbb{R}$ that satisfies the following two properties:

**Capacity Constraints:** For all $u, v \in V$, we require $0 \leq f(u, v) \leq c(u, v)$
**Flow Conservation:** For all $u \in V - \{s, t\}$, we require

$$\sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v) \tag{1}$$

When $(u, v) \notin E$, there can be no flow from $u$ to $v$, and $f(u, v) = 0$. The value $|f|$ of a flow $f$ is defined as

$$|f| = \sum_{v \in V} f(s, u) - \sum_{v \in V} f(v, s) \tag{2}$$

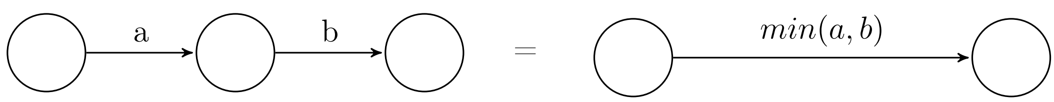
### 1.0.2 An example of a flow

For example the transporting example from the book. A firm wants to transport goods from 1 place to anther, using a third part driver.

Guessing this particular section is not thaaaaat important for the exam. ;o)

### 1.0.3 Modelling problems with antiparallel edges

Antiparallel edges is 2 edges going to/from $(v, u)$ - so 2 edges between 2 vertices but with opposite directions. To come around that, we transform our network into an equivalent one containing no antiparallel edges (adding an extra vertex for that, so we can split one of the edges). The resulting network is equivalent to the original one, due to the fact, that you do not add or subtract anything from the capacity. It is the same:



### 1.0.4 Networks with multiple sources and sinks

A maximum flow problem may have several sources and sinks, rather than just one of each. To fix that, we just add a supersource and a supersink with infinity capacity from $s$ to each of the multiple sources.

## 1.1 The Ford-Fulkerson Method

```
FORD-FULKERSON-METHOD(G, s, t)
initialise flow f to 0
while there exists an augmenting path p in the residual network G_f do
    augment flow f along p
end while
return f
```

### 1.1.1 Residual Network

Suppose that we have a flow network $G = (V, E)$ with source $s$ and sink $t$. Let $f$ be a flow in $G$, and consider a pair of vertices $u, v \in V$. We define the residual capacity $c_f(u, v)$ by

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E, \\ f(v,u) & \text{if } (v,u) \in E, \\ 0 & \text{otherwise} \end{cases}$$

### 1.1.2 Augmenting paths

### 1.1.3 Cuts of Flow Networks

### 1.1.4 The basic Ford-Fulkerson Algorithm

### 1.1.5 Analysis of Ford-Fulkerson

### 1.1.6 The Edmond-Karp Algorithm

## 1.2 Maximum bipartite matching

### 1.2.1 The Maximum-bipartite-mathing problem

### 1.2.2 Finding a maximum bipartite matching

# 2 Fibonacci Heaps

# 3 NP-Completeness

# 4 Randomised Algorithms

# 5 Exact Exponential Algorithms

# 6 Approximation Algorithms

# 7 Computational Geometry

# 8 Linear Programming and Optimisation