

Alexander Kröller
Henning Hasemann
Stephan Friedrichs

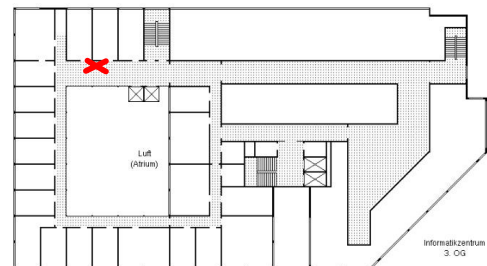
Verteilte Algorithmen Übung 4 vom 19. 6. 2012

Abgaben zu A und T am Dienstag, dem
27. 6. 2012, entweder

- vor der Vorlesung im IZ358, oder
- bis 9:40 im Hausaufgabenrückgabeschrank.

Bitte die Blätter vorne deutlich mit eigenem Namen versehen!

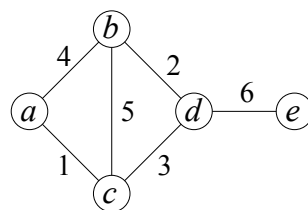
Abgaben zu P zum 15. 7. per Mail an
hasemann@ibr.cs.tu-bs.de.



A — Allgemeiner Teil

Diese Aufgaben können von jedem bearbeitet werden, egal ob sich ansonsten für T oder P entschieden wird.

Aufgabe A1: Beschreibe den Ablauf von GHS (Alg. 2.18) auf diesem Netzwerk, in SYNC:



Gib dazu für jede Kommunikationsrunde an, über welche Kanten welche Nachrichten geschickt werden, und welchen Inhalt diese Nachrichten haben. **(2 P.)**

Aufgabe A2: Gib für jedes $n \in \mathbb{N}$ ein zusammenhängendes Netzwerk mit der folgenden Eigenschaft an: In GHS (2.18), in SYNC, durchläuft jeder Knoten nur eine Leveländerung (jeder Knoten v speichert $L(F_i)$ für das Fragment F_i mit $v \in F_i$, eine Leveländerung ist also, wenn sich diese gespeicherte Zahl ändert.)

Gib für dein Netzwerk den Ablauf von GHS an, um diese Eigenschaft zu belegen.

(2 P.)

T — Theoretischer Track

Diese Aufgabe schliesst sich mit P aus:

Aufgabe T1: Entwirf einen Algorithmus, der in SYNC und CONGEST (d.h., in jeder Runde kann jeder Knoten jedem seiner Nachbarn genau eine Nachricht der Grösse $\mathcal{O}(\log n)$ schicken) einen MST in Zeit $\mathcal{O}(\log n)$ bestimmt — allerdings für Graphen G mit Durchmesser $\text{diam}(G) = 1$, sprich vollständige Graphen. **(6 P.)**

P — Praktischer Track

Diese Aufgabe schliesst sich mit T aus. Sie läuft über **Blatt 4** und **Blatt 5** zusammen, ihr habt Zeit bis zum 15. 7.

Aufgabe P1: Entwirf und implementiere (in der Wiselib, klar) einen Algorithmus für kürzeste Wege in allgemeinen gewichteten Graphen, mit IDs. Ein Knoten dient als Quelle (typischerweise ist dies ein Knoten mit Aussen-, d.h. Internetanbindung). Die Gewichte werden von einer anderen Komponente gestellt, sie stellen bspw. Linkqualitäten dar. Der Algorithmus soll mit dynamischen Netzen umgehen können, d.h.:

- Einige Knoten starten später („kommen dazu“),
- Knoten können sich beenden,
- Knoten können sich bewegen und darüber ihre Nachbarschaft ändern,
- Durch die Bewegungen kann das Netzwerk geteilt bzw. wieder vereint werden.
- Die Gewichte der Kanten können sich ändern.

Euer Algorithmus soll diesmal auch auf eingebetteten Systemen laufen können. Daher gibt es zwei Teilaufgaben:

- a) Entwirf und implementiere einen Algorithmus in der Wiselib; zur Abgabe gehört auch eine Beschreibung des Algorithmus, eine Begründung seiner Korrektheit. Dafür gibt es bis zu **6 P.**
- b) Implementiere den Algorithmus so, dass er auch auf den iSense-Sensorknoten des Instituts läuft. Dafür gibt es bis zu weitere **6 P.**

In der **grossen Übung am 26.6.** werden Details zur Implementierung und zum Testen besprochen. **(6+6 P.)**