Fachbereich Informatik
Wesley Terpstra
Dr.-Ing. Ilia Petrov
Prof. Alejandro Buchmann, Ph. D.
Sommersemester 2009

TECHNISCHE UNIVERSITÄT DARMSTADT

Grundlagen der Informatik II

3. Praktikumsaufgabe

Der vorangegangene Einsatz war ein solcher Erfolg, dass die Dienststelle beschlossen hat, die Operation weiter zu eskalieren. Neu gewonnene Erkenntnisse ermöglichen einen detaillierten Überblick über die Geldwäschetätigkeiten unserer Feinde. Mehrere Tarnfirmen dienen den Kriminellen zur Einzahlung des illegalen Geldes. Die Mittel werden dann im Rahmen von legalen Interaktionen durch ein Netz von Firmen geschleusst. Abschließend werden die Mittel, deren krimineller Ursprung verschleiert wurde, wieder entnommen und für weitere Aktivitäten der Kriminellen eingesetzt. Die Dienststelle plant, den Geldfluss von der Eingabe zur Ausgabe vollständig zu stoppen und somit das kriminelle Netzwerk trocken zu legen.

Leider sind einige Firmen, die im Geldwäschenetzwerk missbraucht werden, nicht kriminell, sondern bilden einen wichtigen Teil unserer Wirtschaft. Daher können nicht einfach alle beteiligten Firmen geschlossen werden.

Die Dienststelle hat Sie damit beauftragt, das Gesamtsystem zu analysieren und die Geldflüsse zu identifizieren, mit denen die illegale Geldwäsche vollständig gestoppt werden kann, ohne die restliche Wirtschaft unnötig zu beschädigen. Ihr Algorithmus wird eine Listen von Eingabe- und Entnahmefirmen erhalten. Außerdem haben unsere Wirtschaftsexperten die Geldflüsse zwischen allen beteiligten Firmen identifiziert.

Das Geldwäschenetzwerk lahm zu legen und dabei den Schaden für das Wirtschaftssystem zu begrenzen ist Ihnen als Minimalschnittproblem wohl bekannt. Sie erinnern sich aus Ihrer Ausbildung daran, dass der minimale Schnitt eines Graphen dem maximalen Fluss entspricht.

Zu implementieren: FundingNetwork.java um die Ziele zu identifizieren!

Die Lösung ist bis zum **07. Juni 2009 23:59** einzureichen!

Achtung: Diese Anweisungen sind mit besonderer Sorgfalt zu behandeln und werden sich automatisch selbst zerstören!

Schritt 0: Laden Sie die Sourcecode-Vorgaben von `https://www.dvs.tu-darmstadt.de/teaching/inf2/2009-de/Problem3.zip` herunter.

Schritt 1: Implementieren Sie Breitensuche (BFS).

Hinweis: Die Klasse LinkedList unterstützt removeFirst() und addLast() und kann als Warteschlange benutzt.

Schritt 2: Finden Sie einen flussvergrößernden Pfad (augmenting path) mithilfe von BFS.

Hinweis: Suchen Sie dabei mit der Elternrelation einen Pfad von der Senke zur Quelle.

Schritt 3a: Implementieren Sie den Edmonds-Karp Maximum-Flow-Algorithmus.

Hinweis: Zunächst berechnen Sie den Fluss einen flussvergrößernden Pfads.

Hinweis: Passen Sie die Kapazitäten der Kanten im Pfad in symmetrischer Weise an.

Schritt 3b: Fassen Sie mehrere Quellen und Senken eines Graphen mit addSuperSource-Sink zusammen.

Schritt 3: Implementieren Sie den minimalen Schnitt.

Hinweis: Führen Sie Edmonds-Karp aus, um den Residualgraphen zu erhalten.

Hinweis: Finden Sie alle Knoten im Residualgraphen, die von der Quelle erreichbar sind.

Hinweis: Alle Kanten von den erreichbaren Knoten zu den nicht erreichbaren Knoten wurden vom maximalen Fluss vollständig genutzt und bilden einen minimalen Schnitt.

Zusätzliche Informationen über dieses Problem (minimaler Schnitt) finden Sie unter:

a) `http://en.wikipedia.org/wiki/Maximum_flow_problem`

b) `http://en.wikipedia.org/wiki/Max-flow_min-cut_theorem`

c) `http://something.i.have.forgot/`

d) Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. *Introduction to Algorithms*, 2nd edition. Chapter 26.2.