Karlsruher Institut für Technologie
Institut für Theoretische Informatik

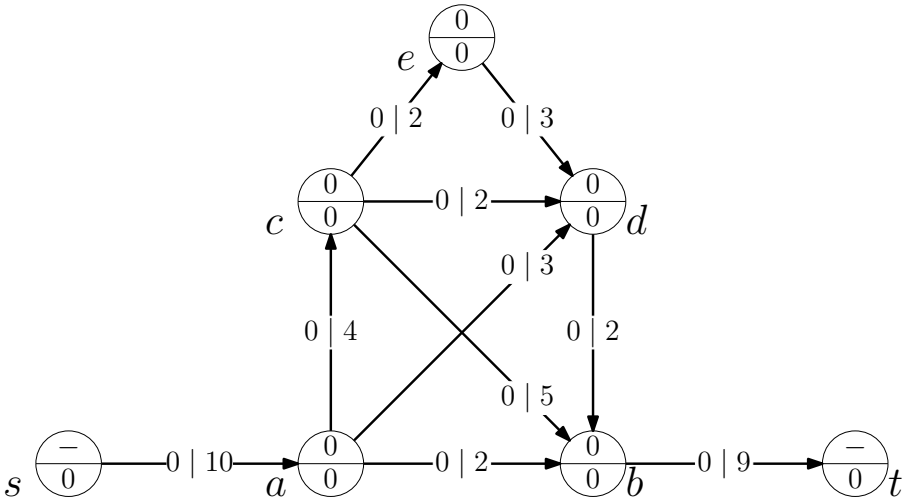
Prof. Dr. Peter Sanders
Moritz Laupichler
Hans-Peter Lehmann

3. Übungsblatt zu Algorithmen II im WS 2022/2023

http://algo2.iti.kit.edu/AlgorithmenII_WS22.php
{sanders, moritz.laupichler, hans-peter.lehmann}@kit.edu

Aufgabe 1 (*Rechnen: preflow-push Algorithmus*)

Gegeben sei folgender Flussgraph:



(das Haus vom Nikolaus mit Vorgarten und Hinterhof)

Knotenbeschriftung: Level (unten), Überschuss (oben)
Kantenbeschriftung: Fluss (vorne), Kapazität (hinten)

Bestimmen Sie den maximalen Fluss von s nach t mit dem generischen *preflow-push* Algorithmus.

Aufgabe 2 (*Analyse: Eigenschaften von Flüssen*)

a) Seien (S, T) und (S', T') zwei minimale (s, t) Schnitte in einem Flußgraphen G. Zeigen oder widerlegen Sie, dass $(S \cup S', T \cap T')$ und $(S \cap S', T \cup T')$ auch minimale (s, t) Schnitte sind.

b) Sei (S, T) ein minimaler (s, t) Schnitt in einem Flussgraphen G. Zeigen oder widerlegen Sie, dass (S, T) ein minimaler (x, y) Schnitt ist f.a. $(x, y) \in S \times T$.

c) Zeigen Sie, dass für den *preflow-push Algorithmus* aus der Vorlesung mit beliebiger Wahl des nächsten Knotens $\mathcal{O}(n^2)$ die bestmögliche obere Schranke ist.

Aufgabe 3 (*Analyse: Matchings*)

a) Zeigen oder widerlegen Sie. Existiert in einem bipartiten Matching ein maximaler alternierender Pfad, dessen erste und letzte Kante nicht Teil des Matchings sind, so hat das zugehörige Flussnetzwerk einen augmentierenden Pfad.

b) Gegeben ein bipartiter Graph $G = (L \cup R, E)$ mit $|L| = |R| = n$. Bezeichne $neigh(S) \subseteq R$ die Nachbarn der Knoten aus $S \subseteq L$. Zeigen Sie, gdw. ein perfektes Matching für G existiert, gilt für alle $S \subseteq L$: $|neigh(S)| \geq |S|$.

c) An einem Tanzkurs wollen n Männer und n Frauen teilnehmen. Jeder Teilnehmer kennt genau k Teilnehmer des anderen Geschlechts (alle Bekanntschaften sind beidseitig). Ist es möglich eine Paarung zu finden, in der jeder mit einem Bekannten tanzt?

   Der Tanzkurs dauert genau k Wochen. Ist es möglich für jede Woche Paarungen zu bilden, so dass immer zwei Bekannte zusammen tanzen, sich aber keine Paarung wiederholt?

Aufgabe 4 (*Entwurf+Analyse: Qualitätskontrolle)*)

Sie sind beauftragt worden, einen Algorithmus für die Abteilung zur Qualitätskontrolle zu entwerfen, der die Validierung der wöchentlichen Resultate übernimmt.
Am Ende jeder Woche erhalten Sie dafür eine Liste L der in dieser Woche produzierten Bauteile mit den drei Angaben: (Typ; ID des Bauteils selbst; ID des Bauteils, in dem es verbaut wurde). Zudem erhalten Sie eine Liste D mit den IDs aller Bauelemente, die in derselben Woche von der Qualitätskontrolle als defekt markiert worden sind. Beide Listen sind potentiell zu groß für den Hauptspeicher und enthalten die Daten in unsortierter Reihenfolge.
Ihre Aufgabe ist es zu überprüfen, ob alle Bauteile, die selbst defekte Bauteile enthalten, als defekt markiert worden sind und falls nicht, dies zu korrigieren. Zu Ihrer Übersicht steht Ihnen ein Schema zur Verfügung, aus dem man ablesen kann, welche Bauteiltypen in welchen anderen verbaut werden. Dieses Schema passt in den Hauptspeicher.

a) Erweitern Sie Liste L um die Angabe aus Liste D, ob das jeweilige Bauteil defekt ist. Sie können davon ausgehen, dass diese Information keinen zusätzlichen Speicher benötigt.

b) Definieren Sie eine totale Ordnung $\prec_b$ auf den Bauteilen, die sich für jedes Bauteil aus lokalen Informationen und dem Bauteilschema berechnen lässt. Die Ordnung soll dabei erfüllen, dass in einer nach $\prec_b$ sortierten Liste L jedes Bauteil nach allen in ihm verbauten Bauteilen steht.

c) Verwenden Sie die sortierte und um Angaben zu Defekten erweiterte Liste L, um alle Bauteile zu identifizieren, die defekte Bauteile enthalten.

   Hinweis: Verwalten Sie die als defekt identifizierten aber noch zu betrachtenden Bauteile in einer geeigneten Datenstruktur.

Aufgabe 5 (*Schlechter Zufall*)

Gegeben sei ein randomisierter Algorithmus badBit, der keine Eingabe liest und als Ausgabe zufällig mit Wahrscheinlichkeit p die Zahl 0 und mit Wahrscheinlichkeit $q = 1 - p$ die Zahl 1 liefert. Es sei $0 < p < 1$; der konkrete Wert von p sei aber unbekannt.
Entwerfen Sie einen randomisierten Algorithmus fairBit, der keine Eingabe liest und als Ausgabe immer zufällig eine der Zahlen 0 und 1 mit Wahrscheinlichkeit $1/2$ liefert.
Was können Sie über die Laufzeit Ihres Algorithmus sagen?

Aufgabe 6 (*Analyse: Speicherbandbreite (*)*)

Die Effizienz eines Algorithmus, der auf externem Speicher arbeitet, hängt von der gewählten Blockgröße B in Zusammenspiel mit der maximalen Bandbreite W_{max} und der durchschnittlichen Zugriffszeit T_{seek} des externen Speichers ab.
Bestimmen Sie für die folgenden Fälle die Blockgröße, für die 90% der maximalen Bandbreite ausgereizt werden kann. Sie können davon ausgehen, dass ohne Unterbrechung auf ganze Blöcke in zufälliger Reihenfolge zugegriffen wird. Etwaige Berechnungen können als asynchron angenommen werden. Daher muss für diese keine Zeit berücksichtigt werden.

a) $W_{max} = 144$ MByte$/s$, $T_{seek} = 12ms$ (*Lesen von Festplatte*)

b) $W_{max} = 550$ MByte$/s$, $T_{seek} = 100\mu s$ (*Lesen von SSD*)

c) $W_{max} = 68$ MByte$/s$, $T_{seek} = 60s$ (*Lesen von LTO Streamer*)

Aufgabe 7 (*Entwurf+Analyse: Telekommunikationsgesellschaft*)

Eine Telekommunikationsgesellschaft beauftragt Sie eine Anwendung zu schreiben, die monatlich die k Kunden bestimmt, bei denen sich die Rechnung im Vergleich zum Vormonat am meisten verändert hat. Diese Kunden will sich die Telekommunikationsgesellschaft noch einmal genau anschauen, um ihnen eventuell einen neuen Vertrag anzubieten.
Die zu bearbeitenden Daten werden Ihnen auf (langsamen) Bandspeichern zur Verfügung gestellt. Sie erhalten eine Liste mit den aneinandergefügten Datensätzen jeder Zweigstelle ihres Auftraggebers für den aktuellen Monat. Außerdem haben Sie eine entsprechende Liste für den Vormonat zur Verfügung. Gespeichert sind jeweils Tupel $(Kundennummer, Kosten)$.

a) Geben Sie einen Algorithmus an, der die geforderte Aufgabe erfüllt. Geben Sie außerdem die Laufzeit Ihres Algorithmus an und begründen Sie diese. Sie können davon ausgehen, dass die k zu bestimmenden Kunden in den Hauptspeicher passen.

b) Seien nun die k zu bestimmenden Kunden zu groß, um im Hauptspeicher gehalten zu werden. Ändern Sie Ihren Algorithmus so ab, dass er mit der erhöhten Datenmenge zurecht kommt. Geben Sie die Laufzeit Ihres neuen Algorithmus an und begründen Sie diese.

Ausgabe: 06.12.2022
Besprechung: 20.12.2022