Algorithmen und Datenstrukturen II
SoSe 2024 – Serie 4

1 Activity Selection

Bei einer Veranstaltung werden für die Besucher folgende Aktivitäten geplant:

Aktivität	Startzeit	Endzeit
A	1:00 Uhr	5:00 Uhr
B	2:00 Uhr	3:00 Uhr
C	4:00 Uhr	5:00 Uhr
D	6:00 Uhr	7:00 Uhr
E	7:00 Uhr	9:00 Uhr
F	8:00 Uhr	10:00 Uhr
G	4:00 Uhr	7:00 Uhr
H	8:00 Uhr	9:00 Uhr

a) Angenommen ein Teilnehmer möchte eine maximale Anzahl von Veranstaltungen besuchen, unabhängig deren Länge, und ohne Veranstaltungen verspätet zu erreichen oder verfrüht zu verlassen. Bestimmen Sie eine mögliche Menge von Veranstaltungen, die diesen Wunsch erfüllt mit Greedy Activity Selection. Geben Sie für jede Aktivität die Reihenfolge an, in der sie betrachtet wurde, und ob sie besucht werden kann oder nicht. Ist diese Auswahl eindeutig?

b) Warum ist es ausreichend, dass die Aktivitäten nur nach der Endzeit sortiert werden, jedoch nicht nach einem 2. Kriterium?

Lösung:

Eine mögliche Lösung lautet:

a)

Aktivität	Startzeit	Endzeit	
B	2:00 Uhr	3:00 Uhr	OK
A	1:00 Uhr	5:00 Uhr	inkompatibel
C	4:00 Uhr	5:00 Uhr	OK
D	6:00 Uhr	7:00 Uhr	OK
G	4:00 Uhr	7:00 Uhr	inkompatibel
H	8:00 Uhr	9:00 Uhr	OK
E	7:00 Uhr	9:00 Uhr	inkompatibel
F	8:00 Uhr	10:00 Uhr	inkompatibel

damit könnten Veranstaltungen B,C,D, und H besucht werden. Bei Paaren mit gleichen Endzeiten ist die Reihenfolge der Bearbeitung zufällig. A und C, D und G, sowie H und E könne beliebigt vertauscht werden. Dies kann hier das Ergebnis beeinflussen, da sowohl H als auch E Teil der Lösungsmenge sein könnten. Damit ist auch B,C,D, und E korrekt und die Lösung nicht eindeutig.

b) Die Begründung ist bereits im Beweis der Vorlesung enthalten. Man kann sich diese aber wie folgt bewusster machen. Angenommen es gibt zwei Aktivitäten $a = (x_1, y)$ und $b = (x_2, y)$, die als nächstes an eine Lösungsmenge B angefügt werden können. Wir müssen 3 Fälle unterscheiden:

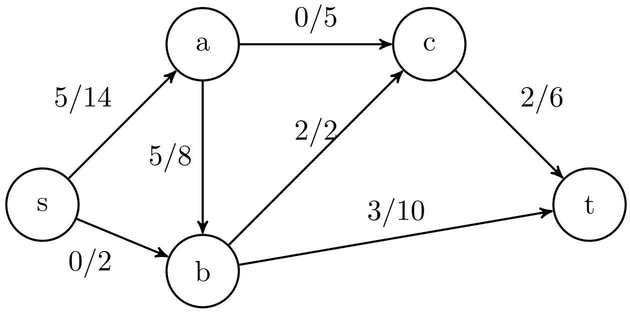
i) a und b sind sind inkompatibel zu B. Die Reihenfolge der Ablehnung ist damit egal.

i) a ist kompatibel und b is inkompatibel zu B (oder andersherum). Die Ablehnung von b ist unabhängig von a und kann somit davor oder danach geschehen.

i) a und b sind kompatibel zu B. Da beide Aktivitäten auf die gleiche Zeit enden, sind sie gegenseitig inkompatibel und nur das jeweils erste wird angefügt. Egal ob a oder b angefügt wird, erweitert sich die Lösung um 1 Element mit identischer Endzeit. Somit verändert sich weder die größe der Lösung, noch folgende Kompatibilitäten.

2 Flussnetzwerke

Gegeben sei folgendes Flussnetzwerk:



a) Geben Sie den Minimal-Cut des Flussnetzwerks an.

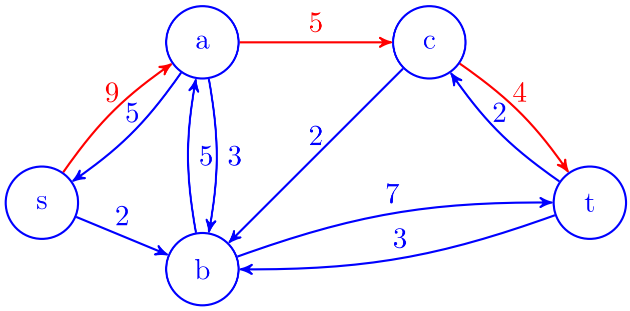
b) Zeichnen Sie den Restgraphen des Flussnetzwerks.

c) Zeichnen Sie in den Restgraphen den Erweiterungspfad mit der größten Restkapazität an. Wie viel Fluss muss danach noch mit weiteren Pfaden erweitert werden?

Lösung:

a) Der minimal-cut ist $(s, b), (a, b), (a, c)$ mit einem Fluss von 15.

b/c)



Es besteht 5 Fluss im Graphen. Der eingezeichnete maximale Pfad erweitert diesen um 4. Aus a) wissen wir, dass ein maximaler Fluss von 15 möglich ist. Damit muss noch ein Rest-Fluss von insgesamt 6 mit weiteren Pfaden gefunden werden.

3 Ford Fulkerson

Geben Sie einen möglichen Graphen an, für den der Ford Fulkerson Algorithmus seinen Worst Case Verhalten entwickeln kann, dass heißt eine maximale Anzahl von Pfaden zum finden des maximalen Flusses benötigt wird.

Universität Leipzig Institut für Informatik Bioinformatik/IZBI	**Algorithmen und Datenstrukturen II** SoSe 2024 – Freiwillige Serie 4		
P.F. Stadler, T. Gatter	Ausgabe am 23.04.2024	Lösung am 30.04.2024	Seite 4/4

Lösung:

Idee: Konstruiere das Flussnetzwerk so, dass jeder Erweiterungspfad immer maximal 1 Fluss findet.