**Team**: 4_4, Claudius Baderschneider und Sven-Ole Fedders

Aufgabtenaufteilung

Aufgaben wurden verteilt bearbeitet

- Kruskal

- Prim Algorithmus / Prim Algorithmus - Fibo-Heap

- Random Generator für die Algorithmen

## Bearbeitungszeitraum

| Name | Datum | Dauer |
|---|---|---|
| Fedders | 22.05.15 | 2 Stunden |
| Fedders | 22.05.15 | 2 Stunden |
| Fedders | 23.05.15 | 2 Stunden |
| Fedders | 23.05.15 | 3 Stunden |
| Fedders | 27.05.15 | 8 Stunden |
| Fedders | 28.05.15 | 4 Stunden |
| Baderschneider | 21.05.15 | 3 Stunden |
| Baderschneider | 22.05.15 | 4 Stunden |
| Baderschneider | 24.05.15 | 3 Stunden |
| Baderschneider | 27.05.15 | 8 Stunden |
| Baderschneider | 28.05.15 | 4 Stunden |

**Akuteller Stand:**

Es können randomisierten Graphen für die Algorithmen erzeugt werden. Für den Kruskal Algorithmus und den Prim Algorithmus.

Es können drei Algorithmen zur Minimum Spanning Tree Findung, genutzt werden.

## *Komponenten:*

Kruskal:

Der Algorithmus wurde kleinteilig implementiert sowie Aufteilung in Hilfsfunktionen ansonsten Äquivalent zu der Beschreibung aus der Vorlesung.

Prim-Algorithmus:

Es wird eine Wrapper-Klasse genutzt um die Daten zu analysieren und zu sortieren.

Ansonsten nutzt der Algorithmus eine PriorityQueue die einmal mit einem Comparator und einmal mit einem Fibonacci-Heap arbeitet.

Bei der Comparator Queue wird für die Wrapper Klasse ein Comparator angelegt, der die Distanz mit compareTo vergleicht und so die Priorität der Warteschlange festlegt.

Beim Fibonacci-Heap wird ein Schlüsselwert festgelegt, der dafür sorgt, dass die Elemente so mit höchster Priorität entnommen werden können.

Am Anfang des Algorithmus wird ein eine temporäre Liste mit der WrapperKlasse als Typ erzeugt und Initialisierungs-Werten befüllt. Dies dient dazu, dass während die Schleife des Algorithmus erkannt werden kann, ob es eine günstigere Verbindung gibt und ob ich diese Knotenpunkt schon abschließend betrachtet habe.

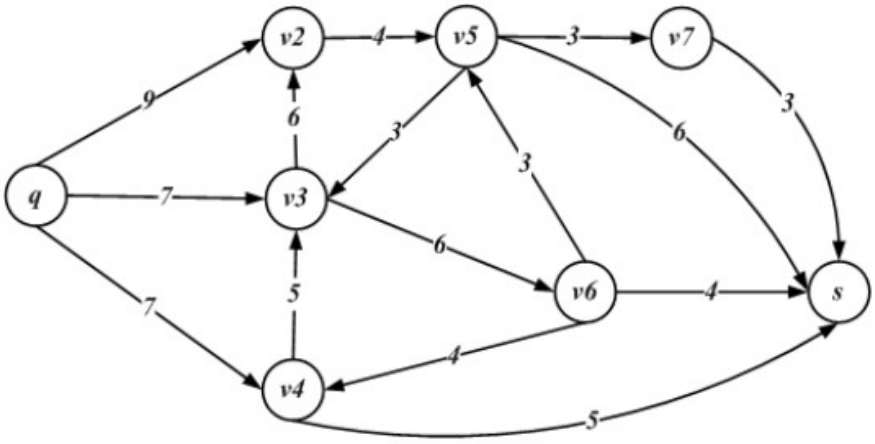
Graph-Generator:

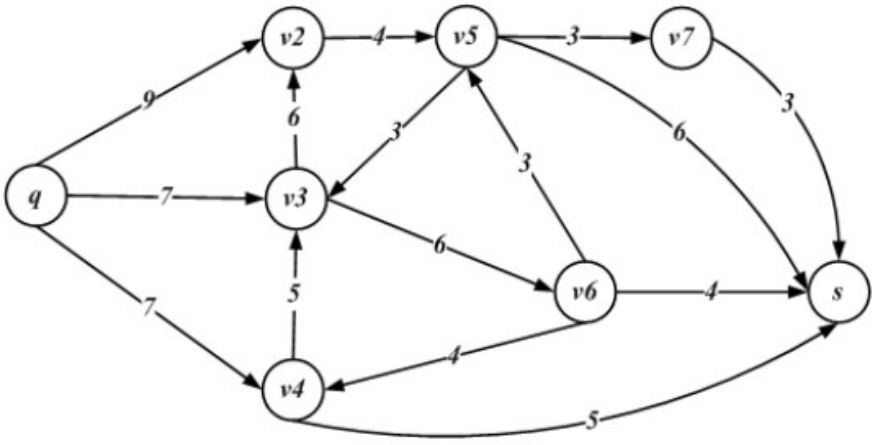
Der Random-Graph Generator gibt einen beliebig großen Gerichteten und Gewichteten endlichen Graphen zurück.

Dieser Graph ist ein zusammenhängender Graph ohne Schleifen.

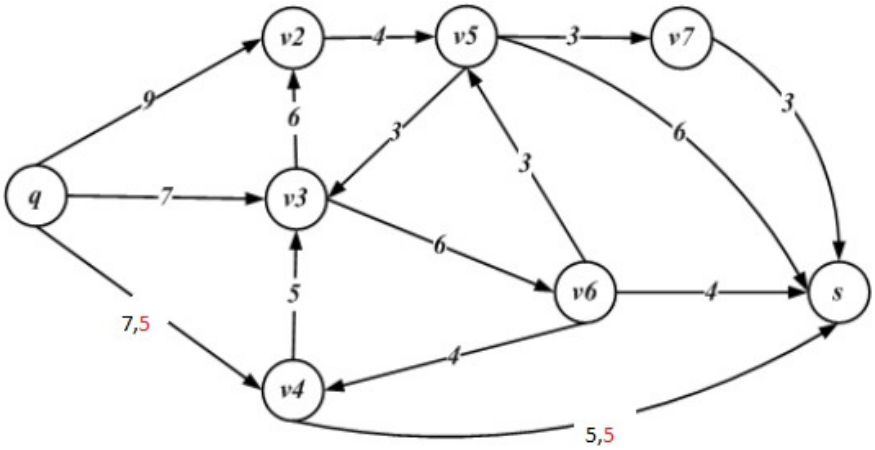
Theorieteil:

Aufgabe IX:

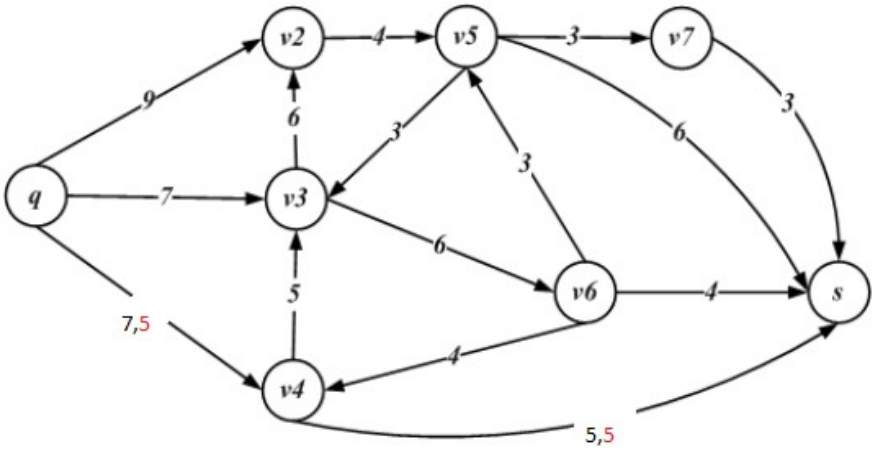


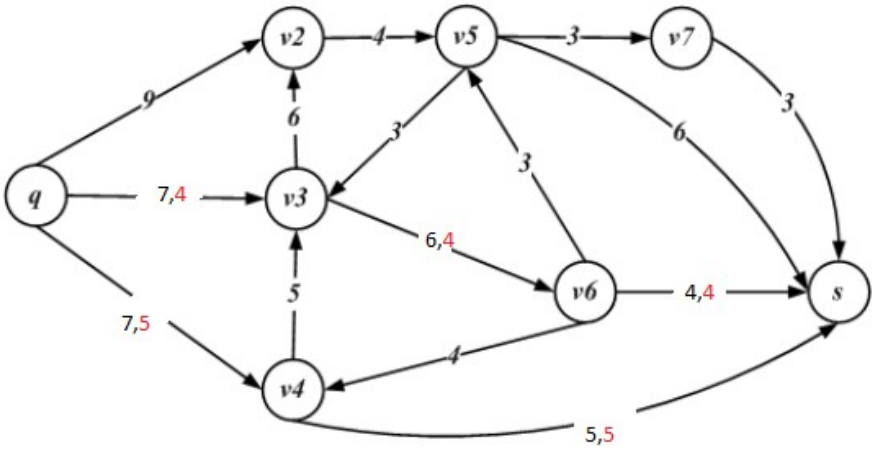

| Knoten | q | v4 | s |
|---|---|---|---|
| Kennung | (/,oo) | (+q,7) | (+v4,5) |



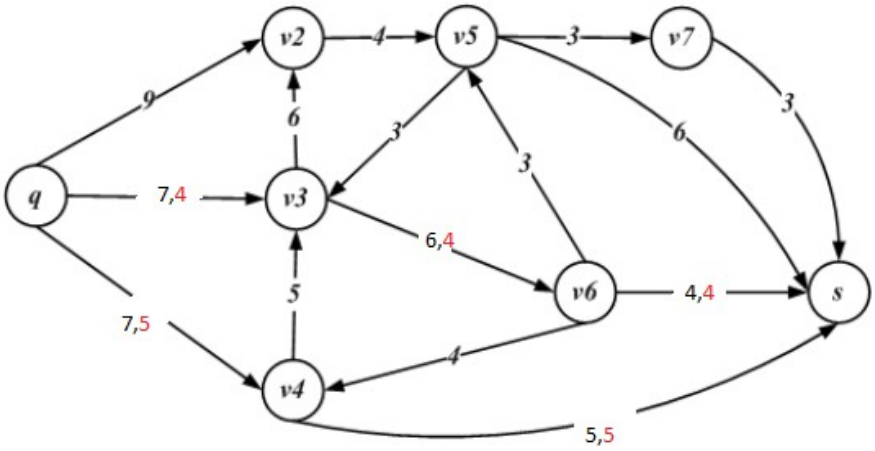

| Knoten | q | v4 | s |
|---|---|---|---|
| Kennung | (/,oo) | (+q,7) | (+v4,5) |



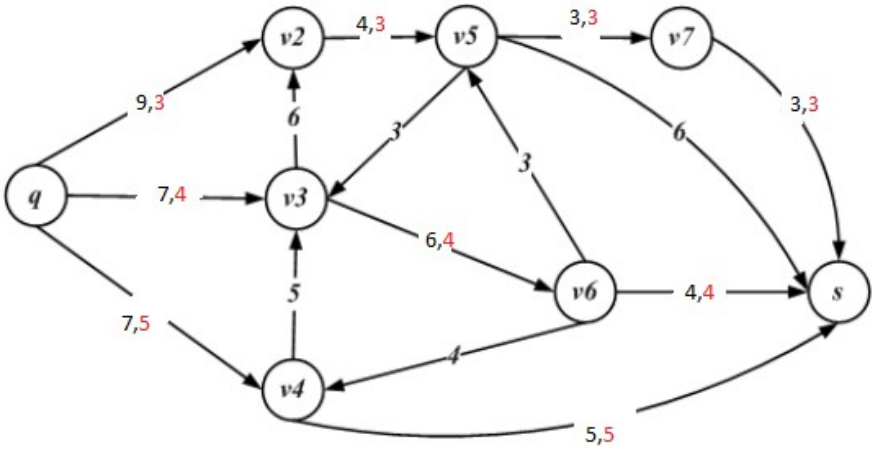

| Knoten | q | v3 | v6 | s |
|---|---|---|---|---|
| Kennung | (/,oo) | (+q,7) | (+v3,4) | (+v6,4) |



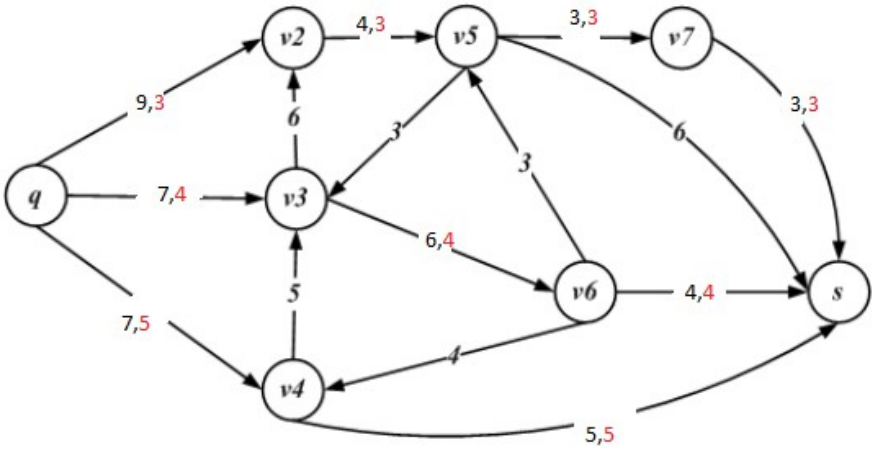

| Knoten | q | v3 | v6 | s |
|---|---|---|---|---|
| Kennung | (/,oo) | (+q,7) | (+v3,4) | (+v6,4) |



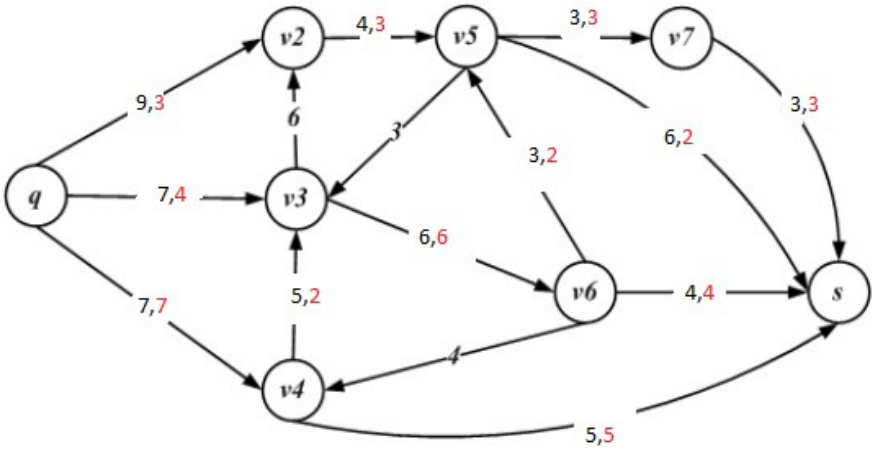

| Knoten | q | v2 | v5 | v7 | s |
|---|---|---|---|---|---|
| Kennung | (/,oo) | (+q,9) | (+v2,3) | (+v5,3) | (+v7,3) |



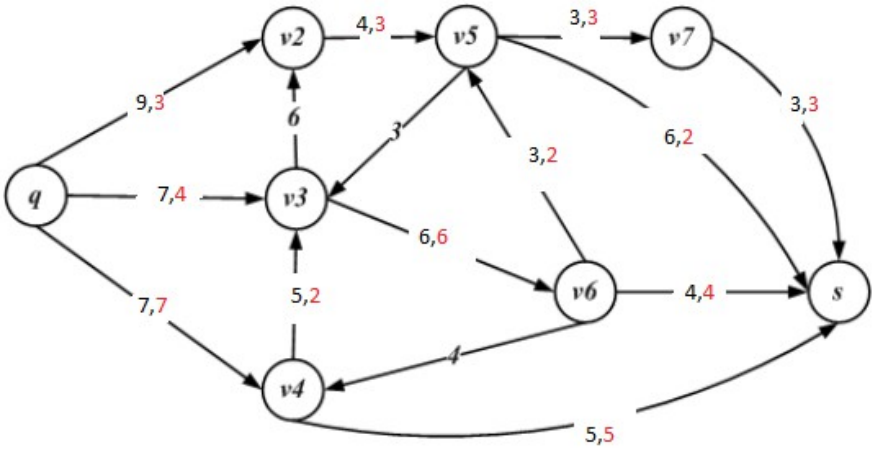

| Knoten | q | v2 | v5 | v7 | s |
|---|---|---|---|---|---|
| Kennung | (/,oo) | (+q,9) | (+v2,3) | (+v5,3) | (+v7,3) |



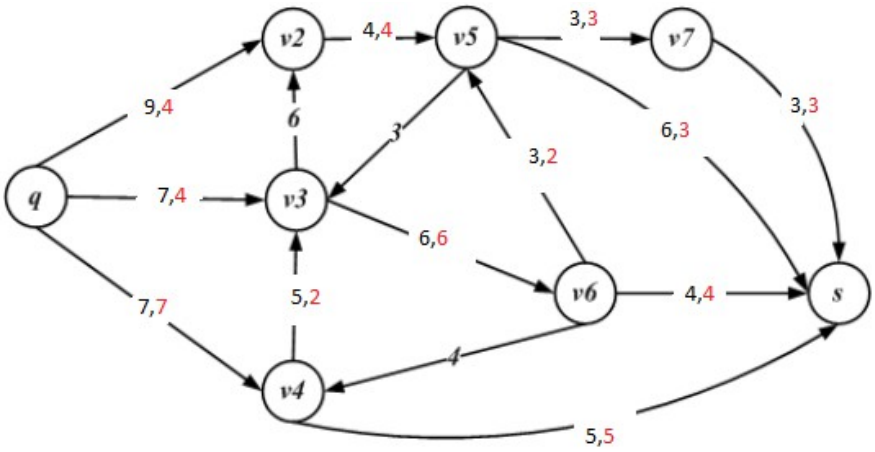

| Knoten | q | v4 | v3 | v6 | v5 | s |
|---|---|---|---|---|---|---|
| Kennung | (/,oo) | (+q,2) | (+v4,2) | (+v3,2) | (+v6,2) | (+v5,2) |




| Knoten | q | v4 | v3 | v6 | v5 | s |
|---|---|---|---|---|---|---|
| Kennung | (/,oo) | (+q,2) | (+v4,2) | (+v3,2) | (+v6,2) | (+v5,2) |




| Knoten | q | v2 | v5 | s |
|---|---|---|---|---|
| Kennung | (/,oo) | (+q,6) | (+v2,1) | (+v5,1) |




| Knoten | q | v2 | v5 | s |
|---|---|---|---|---|
| Kennung | (/,oo) | (+q,6) | (+v2,1) | (+v5,1) |

Der Maximal Fluss beträgt:

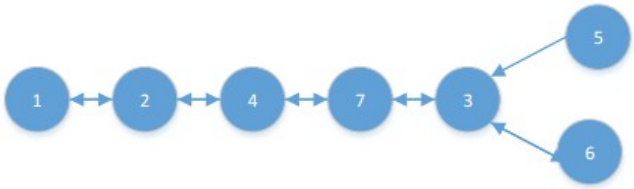
5+4+3+2+1 = 15

Aufgabe VII:

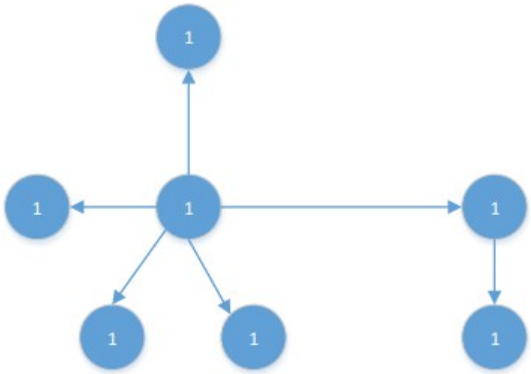
1)

1.


2.




3.



2)