

Prof. Dr. Sándor P. Fekete  
Stephan Friedrichs

## Algorithmen und Datenstrukturen Übung 0b vom 30. 10. 2013

Dieses Blatt dient lediglich der persönlichen Vorbereitung. Es wird nicht abgegeben und geht nicht in die Bewertung ein. Die Aufgaben und ihre Lösungen werden in den kleinen Übungen besprochen.

**Aufgabe 1 (Zusammenhang in Graphen):** Betrachte den Graphen aus Abbildung 1.

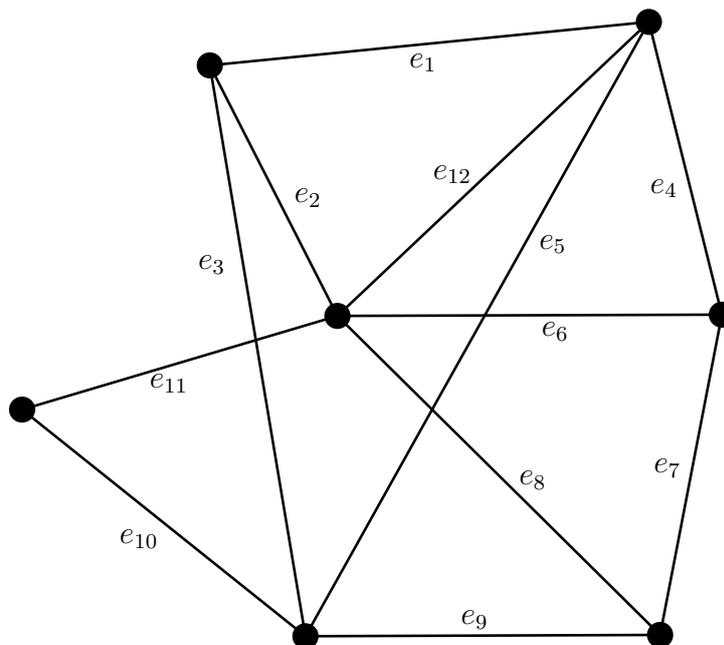


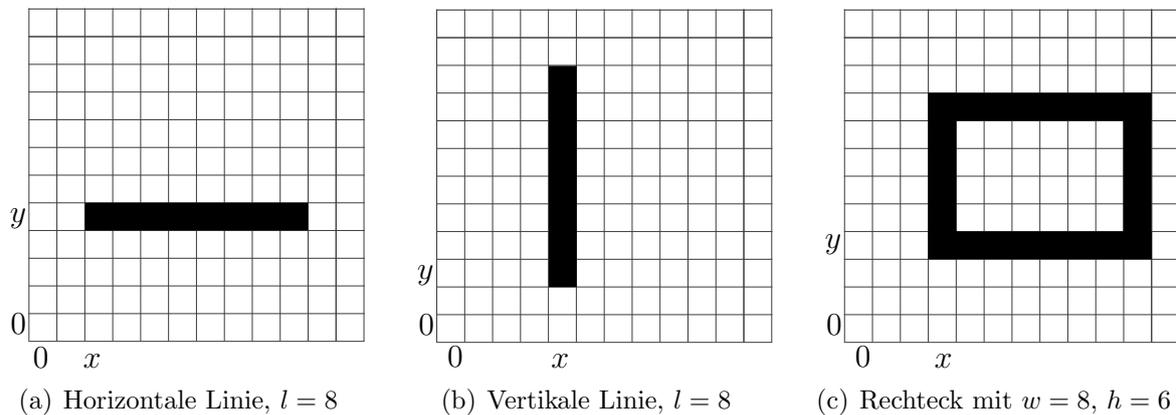
Abbildung 1: Der Graph  $G$

- Wähle möglichst wenige Kanten aus, bis alle Knoten (direkt oder indirekt) verbunden sind, also ein *zusammenhängender Teilgraph* entsteht. Bestimme eine andere Menge von Kanten mit der gleichen Eigenschaft. Wie viele Kanten hast du jeweils ausgewählt?
- Lösche eine möglichst kleine Menge an Kanten, so dass der verbleibende Teilgraph keinen Kreis mehr enthält. Wie viele Kanten bleiben übrig?

**Aufgabe 2 (Zeichnen):** Hier werden Algorithmen zum Zeichnen einfacher Symbole entworfen. Gezeichnet wird in ein Bild  $I$ , auf dessen Pixel mit Koordinaten  $(x, y)$  mit

$I[x, y]$  zugegriffen werden kann. Der Einfachheit halber haben wir nur zwei Farben: Ist  $I[x, y] = 0$ , ist der Pixel  $(x, y)$  weiß, ist  $I[x, y] = 1$ , ist er schwarz. Gesetzt werden kann ein Pixel zum Beispiel mit  $I[3, 7] \leftarrow 1$ . Dabei ist  $(0, 0)$  die untere linke Ecke, siehe Abbildung 2.

Für den Rest der Aufgabe darf angenommen werden, dass sich alle Koordinaten im zulässigen Bereich befinden, dass  $I$  groß genug für die jeweils geforderten Operationen ist und dass anfangs alle Pixel auf weiß gesetzt sind.



**Abbildung 2:** Zeichenroutinen

```
function DRAW_HORIZONTAL_LINE( $I, x, y, l$ )
```

```
end function
```

**Algorithmus 1:** Zeichnen einer horizontalen Linie der Länge  $l$  von  $(x, y)$  nach rechts

- Vervollständige Algorithmus 1. Der soll ausgehend von den Koordinaten  $(x, y)$  eine horizontale Linie der Länge  $l$  nach rechts zeichnen, siehe Abbildung 2(a).
- Vervollständige Algorithmus 2. Diesmal soll ausgehend von  $(x, y)$  eine vertikale Linie mit Länge  $l$  nach oben gezeichnet werden, wie in Abbildung 2(b).
- Implementiere Algorithmus 3. Dieser soll, genau wie in Abbildung 2(c), ein achsenparalleles Rechteck zeichnen. Benutze dabei dabei die Funktionen aus den vorigen Aufgabenteilen!

```
function DRAW_VERTICAL_LINE( $I, x, y, l$ )
```

```
end function
```

**Algorithmus 2:** Zeichnen einer horizontalen Linie der Länge  $l$  von  $(x, y)$  nach oben

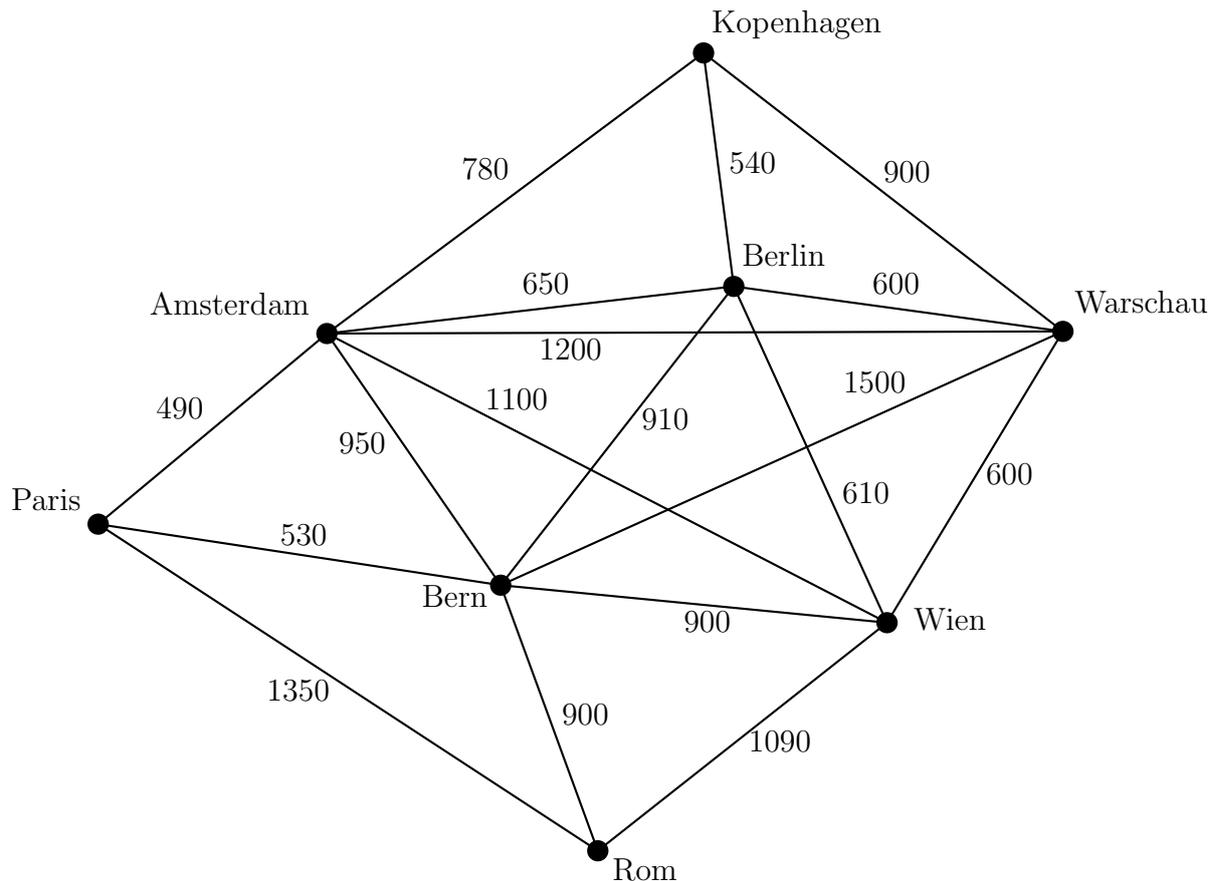
```
function DRAW_RECTANGLE( $I, x, y, w, h$ )
```

```
end function
```

**Algorithmus 3:** Zeichnen eines Rechtecks ab  $(x, y)$  mit Breite  $w$  und Höhe  $h$  nach oben rechts

- d) Entwirf einen Algorithmus, der ein achsenparalleles Rechteck zeichnet, das komplett gefüllt ist. Welche Parameter bekommt der Algorithmus? (Hinweis: Die Funktionen aus den vorigen Aufgabenteilen dürfen nach wie vor benutzt werden.)

**Aufgabe 3 (Rundreise):** Abbildung 3 zeigt schematisch einige Reiseverbindungen zwischen europäischen Städten; die Zahlen an den Kanten beschreiben die Entfernung in Kilometern zwischen den verbundenen Städten. Eine *Tour* startet in einer beliebigen Stadt, besucht alle anderen Städte jeweils genau einmal und kehrt dann zum Ausgangspunkt zurück.



**Abbildung 3:** Europäische Städte

- Wie viele Kanten braucht man für eine Tour? (Hinweis: Diese Frage bezieht sich nur auf die Anzahl, sie ist unabhängig von den zurückgelegten Distanzen.)
- Finde eine möglichst kurze Tour, d. h. eine, für die die Summe der Entfernungen der ausgewählten Kanten möglichst kurz ist.
- Wie gut ist deine Tour? Gib eine Mindestdistanz für *jede mögliche* Städterundreise an und vergleiche ihn mit der Länge der von dir gefundenen Tour!