

PUNKTEVERTEILUNG:

1	2	Σ

Aufgabe (1)

(a) Mengen:

Nationalitäten: { Engländer, Norweger, Spanier, Japaner, Ukrainer }

Haustiere: { Zebra, Schnecken, Hund, Fuchs, Pferd }

Zigaretten: { Parliaments, Altes-Gold, Lucky Strike, Kools, Chesterfield }

Farben: { rot, grün, weiß, gelb, blau }

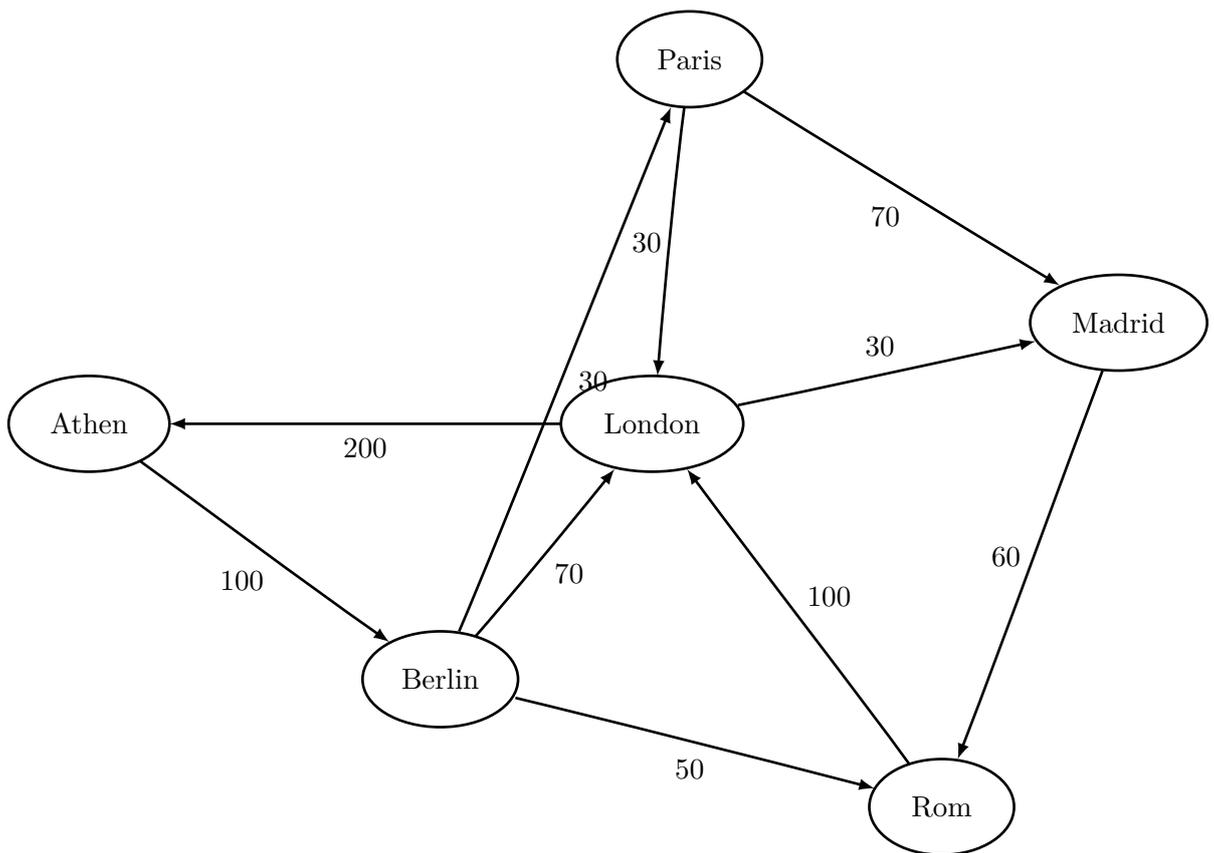
Getränke: { Orangensaft, Tee, Kaffee, Wasser, Milch }

Hausnummern: { 1, 2, 3, 4, 5 }

Prädikate:

imGleichenHaus(x, y): Das Element x ist im selben Haus wie y .*direktRechts*(x, y): Das Element x ist im Haus direkt rechts von dem Haus in dem das Element y ist (also ist die Hausnummer von x gleich der von $y+1$, aber natürlich darf die Hausnummer von y nicht größer als 4 sein).*direktDaneben*(x, y): Das Element x ist im Haus neben dem Haus in dem das Element y ist (also ist die Hausnummer von x entweder der von $y+1$ oder $y-1$, wobei die Hausnummer von x sich zwischen 1 und 5 befinden muss).(b) $\forall h \in \text{Hausnummern} \exists n \in \text{Nation} : \text{imGleichenHaus}(h, n)$ $\forall h \in \text{Hausnummern} \exists t \in \text{Haustiere} : \text{imGleichenHaus}(h, t)$ $\forall h \in \text{Hausnummern} \exists z \in \text{Zigaretten} : \text{imGleichenHaus}(h, z)$ $\forall h \in \text{Hausnummern} \exists f \in \text{Farben} : \text{imGleichenHaus}(h, f)$ $\forall h \in \text{Hausnummern} \exists g \in \text{Getrnke} : \text{imGleichenHaus}(h, g)$ $\forall x, y \in \text{Nation} \forall h \in \text{Hausnummern} : \text{imGleichenHaus}(x, h) \wedge \text{imGleichenHaus}(y, h) \Leftrightarrow (x = y)$ $\forall x, y \in \text{Nation} \forall t \in \text{Haustiere} : \text{imGleichenHaus}(x, t) \wedge \text{imGleichenHaus}(y, t) \Leftrightarrow (x = y)$ $\forall x, y \in \text{Nation} \forall z \in \text{Zigaretten} : \text{imGleichenHaus}(x, z) \wedge \text{imGleichenHaus}(y, z) \Leftrightarrow (x = y)$ $\forall x, y \in \text{Nation} \forall f \in \text{Farben} : \text{imGleichenHaus}(x, f) \wedge \text{imGleichenHaus}(y, f) \Leftrightarrow (x = y)$ $\forall x, y \in \text{Nation} \forall g \in \text{Getraenke} : \text{imGleichenHaus}(x, g) \wedge \text{imGleichenHaus}(y, g) \Leftrightarrow (x = y)$ (c) a) *imGleichenHaus*(Englaender, rot)b) *imGleichenHaus*(Spanier, Hund)

- c) *imGleichenHaus(Kaffe,gruen)*
- d) *imGleichenHaus(Ukrainer,Tee)*
- e) *direktRechts(gruen,weiss)*
- f) *imGleichenHaus(AltesGold,Schnecken)*
- g) *imGleichenHaus(Kools,gelb)*
- h) *imGleichenHaus(Milch,3)*
- i) *imGleichenHaus(Norweger,1)*
- j) *direktDaneben(Chesterfields,Fuchs)*
- k) *direktDaneben(Kools,Pferd)*
- l) *imGleichenHaus(LuckyStrike,Orangensaft)*
- m) *imGleichenHaus(Japaner,Perliaments)*
- n) *direktDaneben(Norweger,blau)*

Aufgabe (2)

- (b) Man geht von der Ursprungsposition aus, in diesem Beispiel ist das Paris. Dort prüft man welche Kanten im Graphen vom Ursprungsknoten ausgehen, in diesem Fall wäre das Madrid und London. Nun kann man den Vorgang des Kantenprüfens auf den gefundenen Knoten nochmals ausführen (quasi rekursiv weiterverfolgen), was zu einer Kaskadenförmigen Rekursion führen kann, bis man irgendwann auf die Ziel-Stadt/Ziel-Knoten trifft. Da dieser Graph Zyklen hat, wäre es möglich in "Schleifen" hin- und herzufliegen (die Rekursion würde Schleifen bilden und würde nie terminieren), muss man verhindern, dass Kanten geprüft werden, die in einer höheren Stelle im Zweig des Rekursionsbaums (der durch die Kaskadenförmige Rekursion beschrieben wird) schon einmal geprüft worden sind. Somit terminiert die Rekursion wenn die Zweige am Ziel angekommen sind oder wenn es keine weiteren Kanten gibt die man verfolgen könnte. Die Zweige die zum Ziel führen, sind die möglichen Flugrouten.

Mögliche Flugrouten wären:

- Paris \rightarrow Madrid \rightarrow Rom
 - Paris \rightarrow London \rightarrow Madrid \rightarrow Rom
 - Paris \rightarrow London \rightarrow Athen \rightarrow Berlin \rightarrow Rom (eher ungewöhnlich, da recht umständlich)
- (c) Das Verfahren von oben lässt sich durchaus wiederholen, indem man alle Routen durchläuft und dann die Preise auf den Kanten zusammenzählt und dort dann die billigste heraussucht. Man kann es sich auch vereinfachen, indem man den aktuell niedrigsten Preis abspeichert, die Zweige absucht und immer die Beträge auf den Kanten zum aktuellen preis aufaddiert. Wenn der aktuelle Preis größer ist als der niedrigste, bereits gefundene Preis ist, kann man abbrechen, ansonsten verfolgt man den Graphen rekursiv weiter bis man dann den Endpunkt erreicht (in diesem Fall setzt man den kleinsten möglichen Preis auf den aktuellen Betrag).
- (d) London \rightarrow Athen \rightarrow Berlin \rightarrow Paris \rightarrow Madrid \rightarrow Rom \rightarrow London

Vorgehen: am Startpunkt London begonnen, von dort die möglichen Alternativen Madrid oder Athen geprüft. Den Weg über Madrid verworfen da es von dort nur nach Rom und dann nach London gehen würde, also den Weg nach Athen eingeschlagen. Dort geht es zwangsläufig nach Berlin weiter. In Berlin hat man die Möglichkeiten London und Paris, wobei aber London direkt verworfen werden kann, da noch nicht alle Städte besucht worden sind, also weiter nach Paris. Dort kann man London auch direkt wieder ausschließen, so dass man weiter nach Madrid fliegt. In Madrid hat man nur noch eine Möglichkeit, Rom und von Rom aus geht es auch nur noch nach London weiter.

Prinzipiell ist der Algorithmus sehr ähnlich zu den vorherigen (mit der Ausnahme dass London der Startpunkt und das Ziel ist und somit zweimal besucht werden darf), jedoch müssen dann die gefundenen Routen darauf geprüft werden, ob jede Stadt einmal angefliegen wurde.