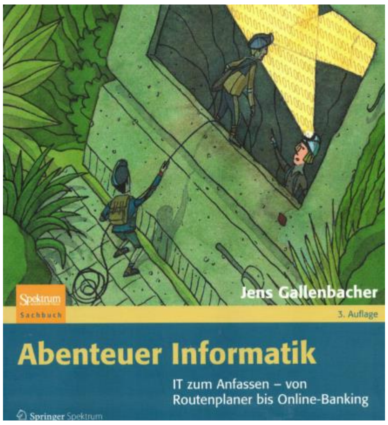
Modul Aktivitätsdiagramm und Algorithmen

Optionaler Auftrag - Kürzeste Wege Algorithmus



Dem nebenstehenden, empfehlenswerten Buch mit den treffenden Titel „Abenteuer Informatik" ist nachfolgendes Beispiel von den Seiten 1 bis 37 in Ausschnitten entnommen.

Einführung

Routenplaner gehören heute schon fast zum Alltag: Viele Autos haben sie bereits eingebaut, wer keinen im Fahrzeug hat, lässt sich den günstigsten Weg zu seinem Ziel oft auf dem heimischen PC ermitteln und druckt ihn aus.

Versuchen wir es doch gleich: Nehmen Sie einen großen Straßenatlas und ermitteln Sie die günstigste Strecke von Stockheim nach Weilheim!

Zu viel Arbeit? Kein Atlas? Also gut – ich hatte in der Einleitung ja versprochen, dass alle notwendigen Materialien hier im Buch seien. Daher arbeiten wir erst einmal mit der folgenden kleinen Welt nach Abbildung 1.1.

Die Karte zeigt Orte, zwei Autobahnen, größere und kleinere Landstraßen. Die roten und blauen Zahlen geben dabei immer die Länge der Straße an, wenn man sie mit dem Auto fährt. Man kann sehen, dass kleine Straßen meistens viel länger sind, als sie scheinen, weil die vielen Kurven und Berge nicht eingezeichnet sind.

Welches ist denn nun der günstigste Weg von Imstadt nach Oppenheim?

Erster Ansatz wäre, zur nächsten Autobahn zu fahren. Aber geht das am besten über Pappstadt oder die Auffahrt Budingen? Außerdem führt ja von Pappstadt aus auch eine direkte Landstraße zum Ziel. Die ist aber wohl gewunden und ziemlich lang. Oder doch lieber die gelbe Straße zum Flughafen und von dort die Autobahn nach Oppenheim nehmen?

Versuchen Sie, die Lösung selbst herauszufinden. Hinweis: Sie müssen 123 Kilometer zurücklegen.

Geschafft? Gut! Dann lehnen Sie sich zurück und genießen Sie den Erfolg.

Wie sind Sie vorgegangen? Sie haben wahrscheinlich alle möglichen Wege durchprobiert und die Entfernung zum Ziel ermittelt. Dann haben Sie sich für den günstigsten Weg entschieden.

Dieses Verfahren gibt es auch bei Computern – es hat sogar einen Namen: die Brute-Force-Methode, also etwa „Brutale Macht". Warum? Weil auf diese Weise etwas größere Probleme nur mit extrem großer Rechenkraft gelöst werden können. Überlegen Sie einmal, wie viele verschiedene Wege Sie schon bei der kleinen

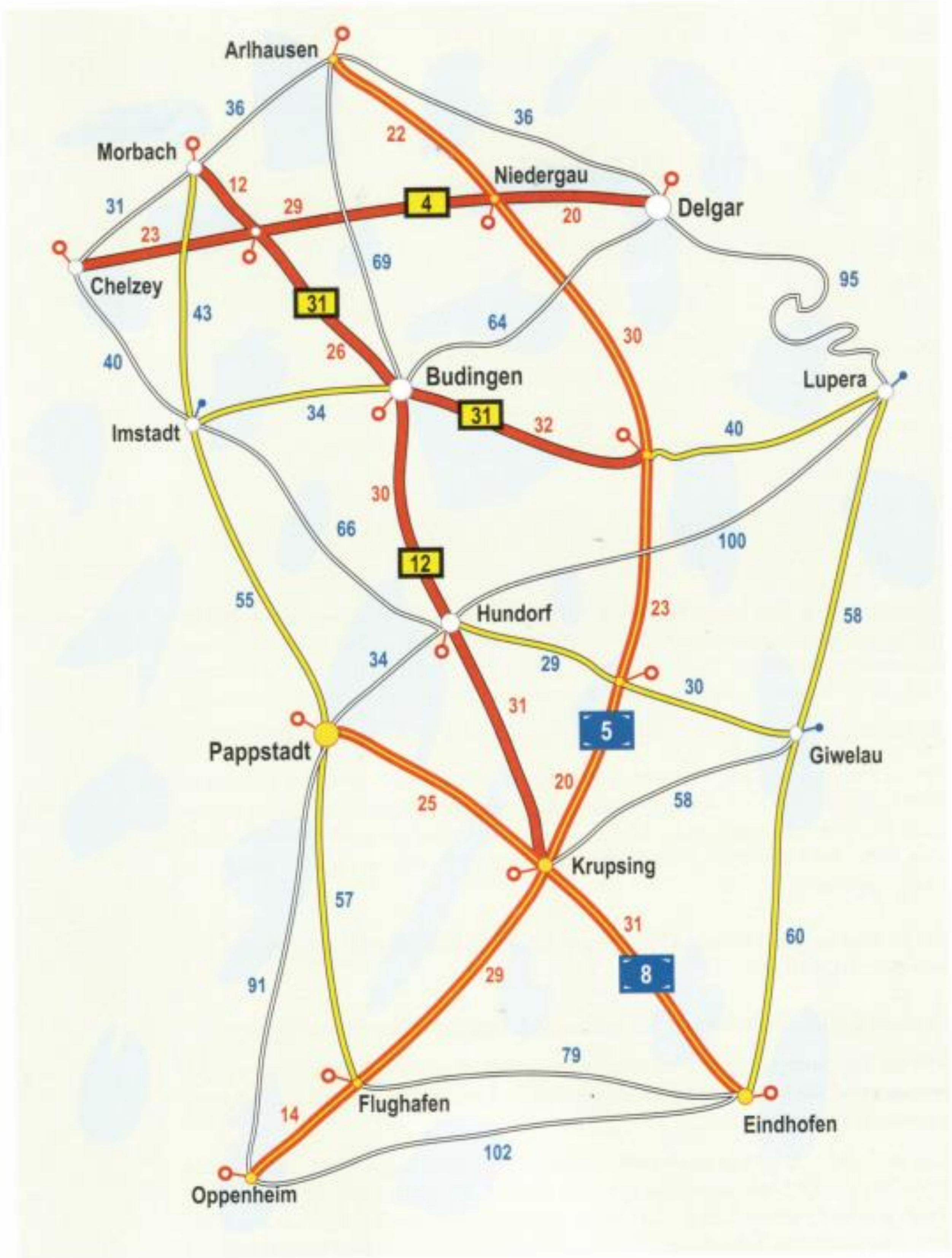


Abbildung 1.1 Die Landkarte aus dem Atlas

Dijkstra und die Ameisen

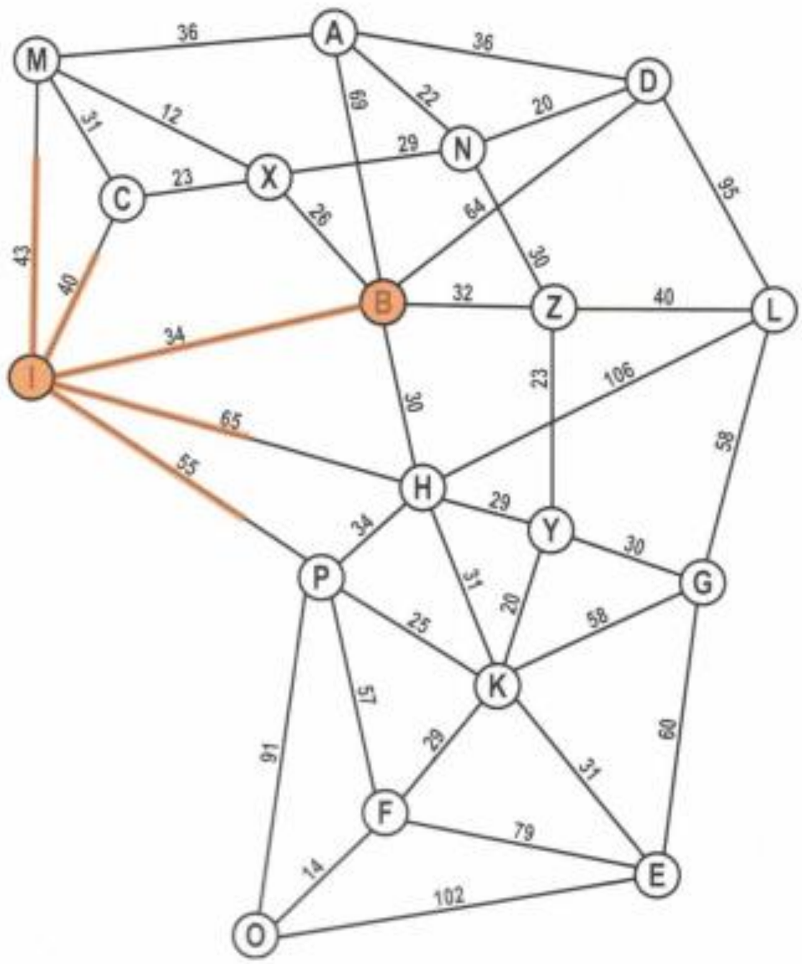
Wie kommen wir denn aber nun zum kürzesten Weg von Imstadt nach Oppenheim?

Der direkte Ansatz, immer ganze Wege zu betrachten, ist ja bereits aufgrund des hohen Aufwands gescheitert. Versuchen wir etwas anderes – vielleicht können wir ja hier von der Natur lernen:

Ein Stamm Ameisen hat auf der Suche nach Futter ein ähnliches Problem: Eine Kundschafterin findet ein großes Stück Fleisch. Welchen Weg sollen die Arbeiterinnen nehmen, um die Beute am schnellsten zu sichern?

Setzen wir also den Stamm Ameisen auf unseren Ausgangspunkt Imstadt bzw. (I). Fünf Wege führen von dort weg, also teilen sich unzählige Ameisen auf, um diese zu erkunden. Wir nehmen jetzt einmal an, dass alle Ameisen gleich schnell sind, zum Beispiel 1 km pro Minute (okay, für unser Problem setzen wir offenbar Turbo-Ameisen ein...).

In der Landkarte verfolgen wir den Weg der Ameisen. Abbildung 1.5 zeigt in Rot ihren Fortschritt nach 34 Minuten.

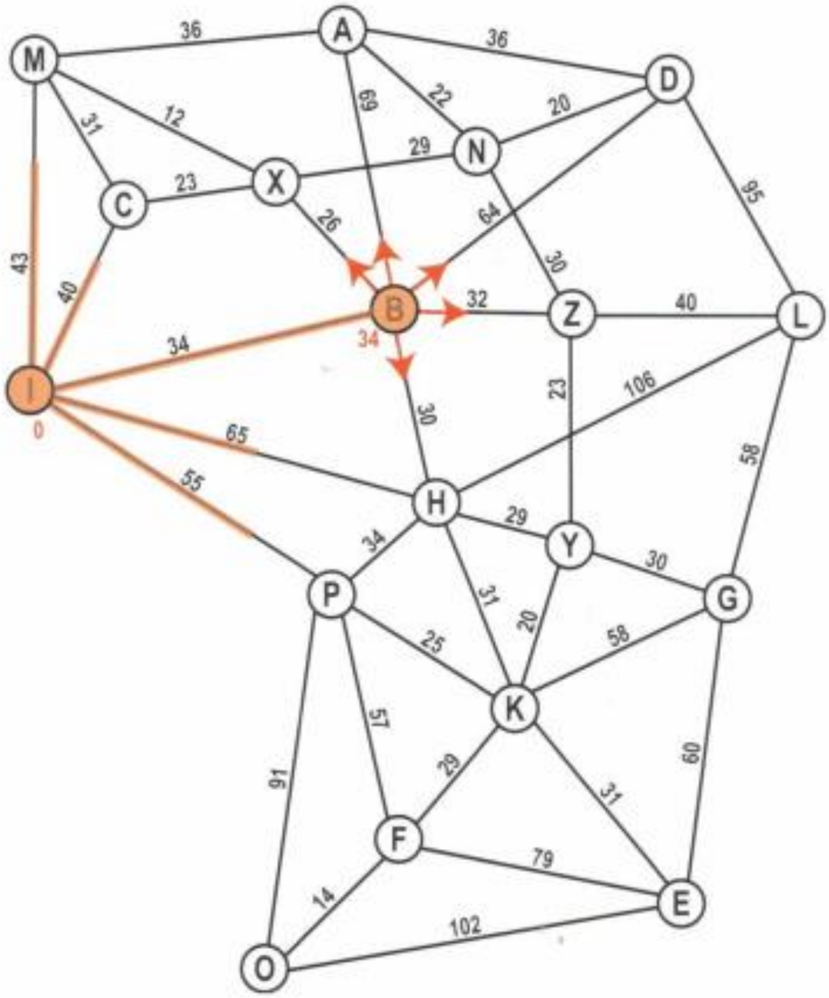


Die Ameisen haben (B) erreicht. Auf den anderen Strecken sind sie noch auf dem Weg. Bitte lassen Sie sich nicht davon täuschen, dass die Strecke mit 34 km am längsten aussieht: Wir verwenden ja explizit keinen maßstabsgetreuen Plan!

Was haben wir dadurch bisher von den Ameisen gelernt?

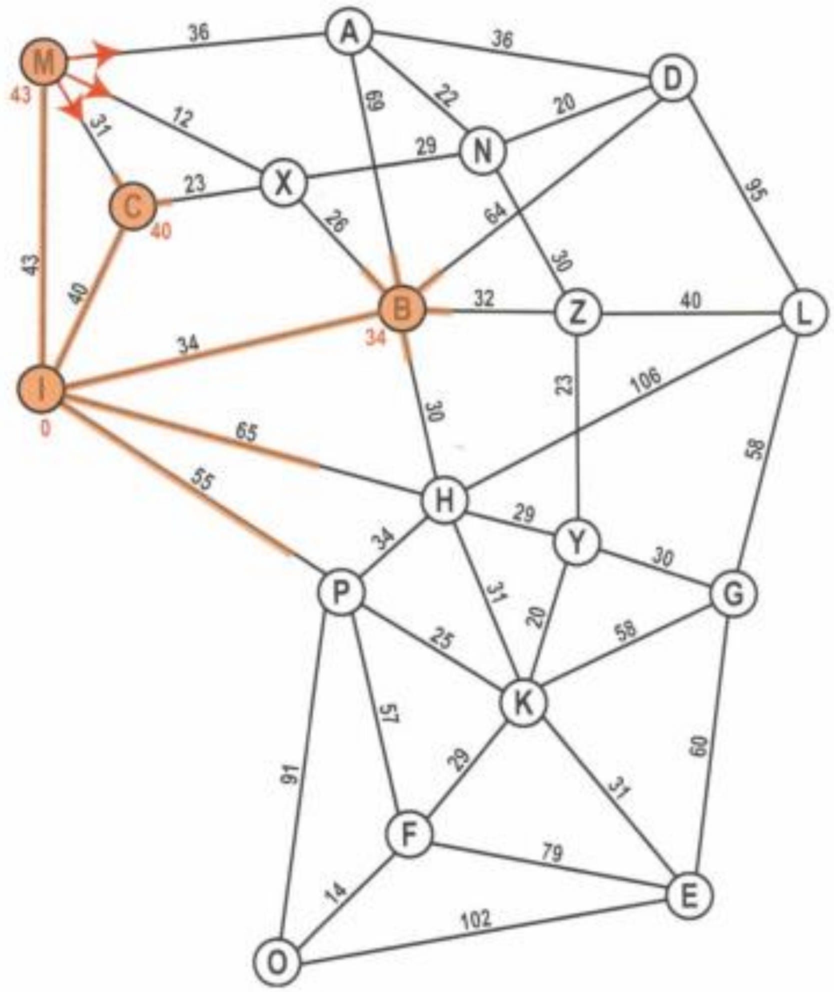
Genau! Um von (I) nach (B) zu kommen, gibt es garantiert keinen günstigeren Weg als den mit 34 km. Die Ameisen haben ja sämtliche bisher für sie möglichen Wege ausprobiert und sind nach 34 km zuerst bei (B) angekommen.

Wie geht es jetzt weiter? Die Ameisen, die bisher nirgendwo angekommen sind, setzen ihren Weg einfach fort. die Ameisen bei B teilen sich erneut auf: Wieder sind fünf Wege möglich. Den bisherigen Erfolg dokumentieren sie, indem sie die bisher zurückgelegte Strecke bei (B) vermerken. Abbildung 1.6 zeigt den Plan der Ameisen.



Nach insgesamt 40 Minuten kommt der nächste Ameisentrupp bei (C) an. Die Insekten sehen, dass sie die Ersten sind, markieren die Strecke, verzeichnen die Anzahl der bisher gelaufenen Kilometer und teilen sich auf die zwei bei (C) weitergehenden Wege auf.

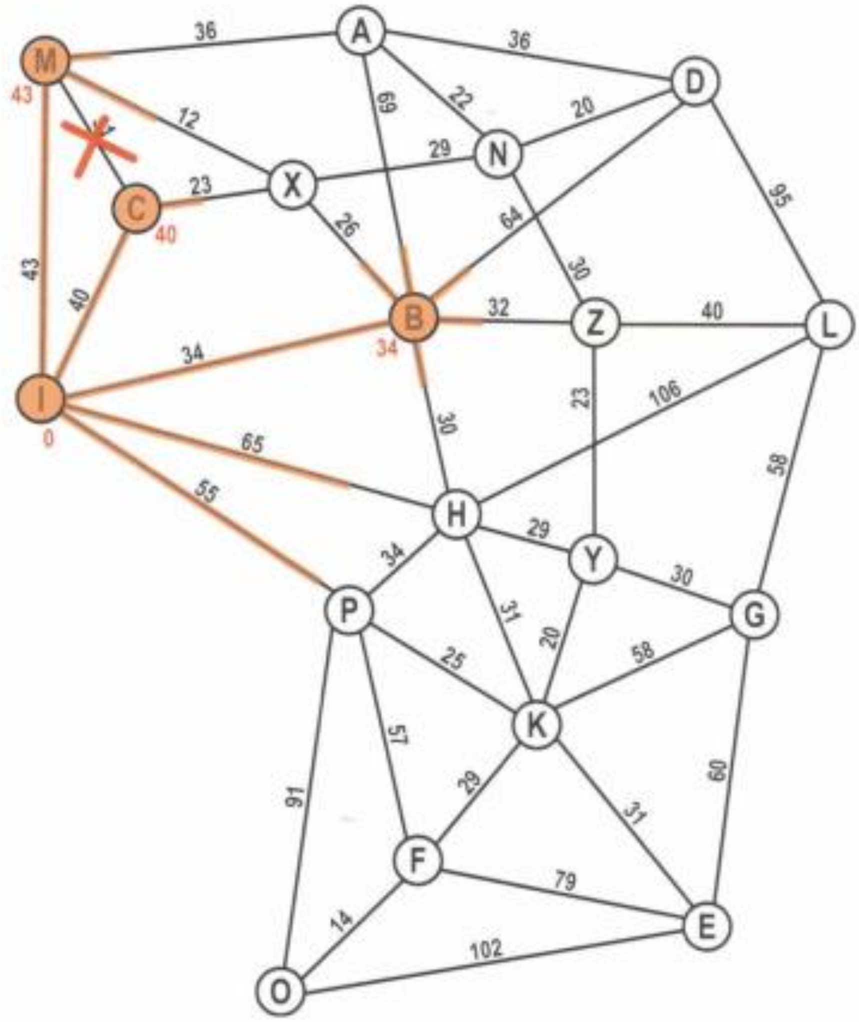
In der 43. Minute kommt dann auch der Trupp bei (M) als Erster an. Auch dieser Weg wird vermerkt (siehe Abbildung 1.7). Von hier aus sind drei weitere Strecken zu erkunden.



Soweit verlief alles nach dem gleichen Schema. Die kürzesten Strecken zu den Städten (B), (C) und (M) stehen nun fest.

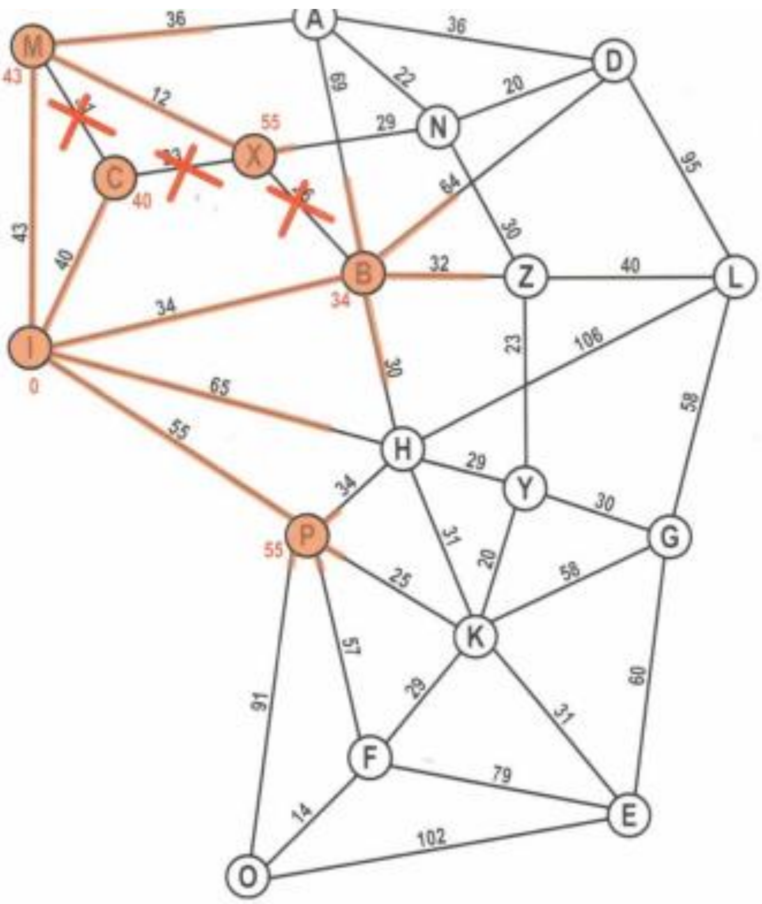
Vielleicht haben Sie bemerkt, dass nun Ameisentrupps sowohl von (M) als auch von (C) ausgehend unterwegs sind – zwischen beiden Städten auf Kollisionskurs. Was passiert jetzt, wenn sie sich irgendwo auf der Strecke dazwischen begegnen? Welche Informationen können sie austauschen? Bringt ihnen das etwas für ihr Ziel, das Gelände zu erkunden?

Genau! Der Trupp von (C) weiß, dass dieses Ziel bereits erreicht ist, die kürzeste Strecke dorthin also feststeht. Der Trupp, der von (M) kommt, kann das Gleiche von seinem Ausgangspunkt berichten. Es ist also sinnlos, weiterzumarschieren. Die Strecke wird als „unbrauchbar" markiert, die Ameisen können wieder zu ihrem Stamm zurück, sie sind sozusagen aus dem Rennen. Abbildung 1.8 zeigt das gleich mit einem dicken roten Kreuz.



Als Nächstes kommen zwei Erkundungstrupps gleichzeitig an: In der 55. Minute erreichen die Ameisen (P) und (X). (P) erreichen sie auf direktem Wege von (I) aus. Bei (X) kommt der Trupp an, der von (M) unterwegs ist (43 km bis (M) plus 12 bis (X)).

Wieder teilen sich die Ameisen auf. Von (X) gibt es nur noch einen erfolgversprechenden Weg, bei den anderen treffen sie recht schnell auf Kameraden und geben die Strecke auf. Die von (P) ausgehenden Touren sind alle noch offen. Abbildung 1.9 zeigt den aktuellen Stand.



Haben Sie das Prinzip verstanden?

Statt immer nur einen Weg auszuprobieren und wieder zu verwerfen, wenn sich ein besserer gefunden hat, erkunden die Ameisen gleichzeitig alle sich bietenden Möglichkeiten.

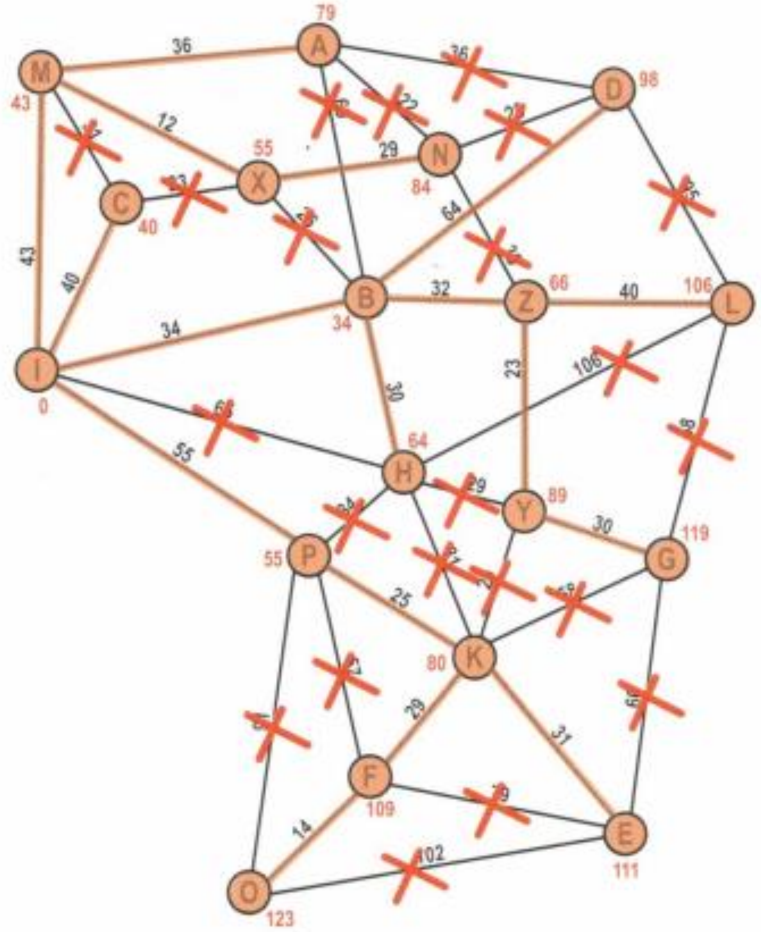
Kommen sie als Erste bei einer Stadt an, wissen sie, dass der genommene Weg der günstigste ist, denn sonst wäre ja ein anderer Erkundungstrupp bereits da (zur Erinnerung: Alle bewegen sich mit der gleichen Geschwindigkeit).

Treffen die Ameisen irgendwo auf Artgenossen, dann wissen sie, dass ihre Reise zu Ende ist, weil sie sich gegenseitig ein „Ich bin schon da" berichten können. Andere haben also das anvisierte Ziel früher erreicht.

Führen Sie zur Übung das Verfahren noch zu Ende und zeichnen Sie den Weg der Ameisen, die gefundenen Strecken sowie verworfene Wege in die Landkarte ein!

Können Sie sich vielleicht gleichzeitig auch schon vorstellen, wie ein Computer das Ameisenprinzip adaptiert?

Die Lösung wird in Abbildung 1.10 dargestellt. Was für Informationen haben wir dadurch jetzt eigentlich gewonnen?



Um von Imstadt zu einem beliebigen anderen Ort zu kommen, folgen Sie dem Pfad der Ameisen:

Von Imstadt nach Oppenheim kommt man so am günstigsten über Pappstadt, Krupsing und Flughafen. Die Gesamtstrecke beträgt 123 km. Ein günstigerer Weg existiert nicht!

Sie haben aber nicht nur die ursprüngliche Aufgabe gelöst, sondern sozusagen als Abfallprodukt noch die kürzesten Wege von Imstadt zu allen weiteren Städten ermittelt. So fährt man zum Beispiel nach Giwelau am besten über Buding und die beiden Autobahnkreuze, die wir mit Z und Y bezeichnet haben.

Beachten Sie bitte auch die „0", die nun der Vollständigkeit halber bei Imstadt steht, denn der Weg von Imstadt nach Imstadt beträgt ja ganz offensichtlich 0 km.

Warum ist das Ameisen-Prinzip für einen Informatiker interessant?

- Es führt in absehbarer Zeit zum Ziel. Da die Ameisen ständig in Bewegung sind und keine Strecke doppelt gehen, müssen sie recht bald alles erkundet haben (maximal nach der Zeit, die dem kürzesten Weg zur am weitesten entfernten Stadt entspricht).

- Es werden immer wieder die gleichen, sehr einfachen Anweisungen benutzt, um die Ameisen zu steuern:
  1. Teile den Trupp auf und folge allen Routen.
  2. Wenn ein Ort erreicht wird: günstigste Strecke dorthin gefunden, weiter bei 1.
  3. Wenn man einem anderen Trupp begegnet: Strecke verwerfen, Ende.

Solche Verfahren lassen sich (normalerweise) gut und einfach für den Computer umsetzen. Eine solche Vorschrift nennt sich dann *Algorithmus.*

Algorithmus
Ein Algorithmus ist eine Handlungsvorschrift zur Lösung eines Problems bzw. einer Kategorie von Problemen. Diese Handlungsvorschriften lassen sich im Allgemeinen in ein Computerprogramm umsetzen. Hierfür müssen sie hinreichend genau formuliert sein.

Wie lösen also unsere Routenplaner im Auto das Problem des kürzesten Weges? Enthalten sie einen Ameisen-Simulator und spielen die genaue Vorgehensweise nach?

Sicherlich wäre das vorteilhafter als die ganz zu Anfang beschriebene „Brute-Force-Methode". Trotzdem ist der Informatiker hier erneut gefragt, das gefundene Verfahren für den Computer zu optimieren.

Können Sie sich vorstellen, wie?

Das Ameisen-Prinzip liegt uns Menschen sehr, weil wir uns gut vorstellen können, wie die kleinen Krabbeltiere die Pfade erkunden. Dadurch erschließt sich auch recht deutlich, warum das Verfahren wirklich die günstigsten Wege hervorbringt.

Wann immer wir so ein „aus dem Leben gegriffenes" Verfahren im Computer benötigen und es daher in Bits und Bytes umsetzen, hilft wieder das Prinzip der Abstraktion:

Überlegen Sie, was vom Ameisen-Prinzip für die Problemlösung relevant ist: Computer benötigen keine Anschauung, diese Anteile sind also zu eliminieren.

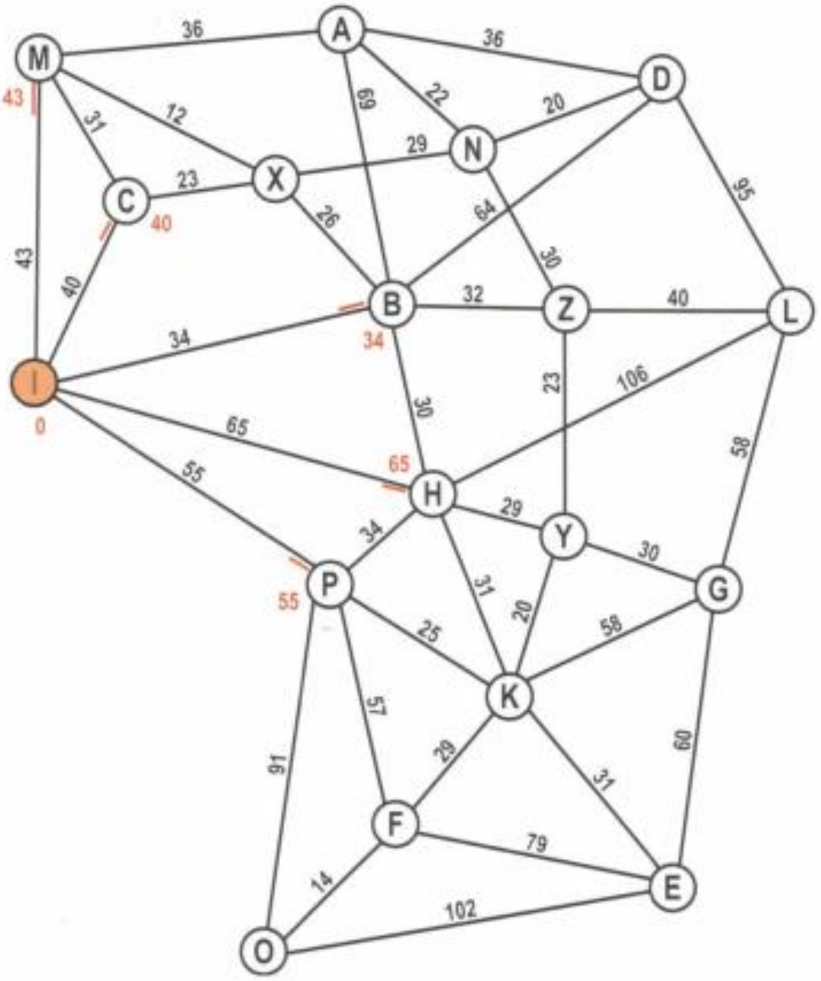
Ob Edsger W. Dijkstra, zuletzt Professor in Texas, 1959 über eine Ameisenstraße lief, ist unbekannt. Er hat jedoch bereits zu diesem Zeitpunkt ein Verfahren vorgestellt, das unser Ameisen-Prinzip im Computer zur Berechnung eines kürzesten Weges nutzt. Es ist bis heute nach ihm benannt.

Doch versuchen Sie zunächst einmal selbst, ein solches Verfahren herauszufinden. Vielleicht kommen Sie ja sogar auf eine bessere Methode, die dann nach Ihnen benannt wird.

Der Dijkstra-Algorithmus

Fangen wir noch einmal ganz von vorne an: Sie sitzen in Imstadt und wollen nach Oppenheim.

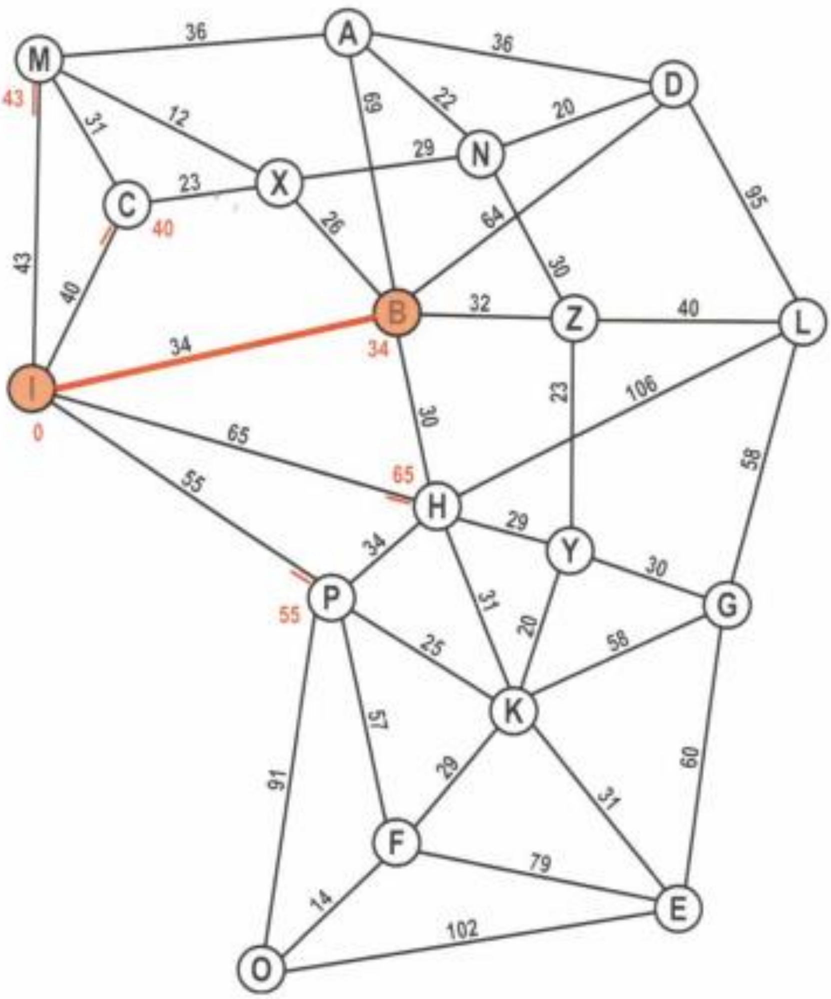
Die Ameisen unseres letzten Experiments liefen nun einfach in alle anderen direkt erreichbaren Städte, um zu ermitteln, wie lange sie dort hin unterwegs sind. Die hierfür notwendigen Zeiten braucht ein Computer jedoch gar nicht zu ermitteln, er liest einfach die Strecken ab. Wie aus Abbildung 1.11 ersichtlich, sind dies ja einfach die verzeichneten Längen der Verbindungslinien zwischen den Städten.



In der Grafik ist außerdem bei allen Zielpunkten markiert, woher Sie kommen, damit die beschriebene Entfernung zutrifft.

Wie ging es mit den Ameisen weiter? Der Trupp, der zuerst bei einer Stadt ankam, markierte die genommene Strecke als „günstig" und verteilte sich auf die umliegenden Strecken.

Für den Computer ist nun gar kein Problem, diese Stadt zu bestimmen: Es ist diejenige, die mit der kleinsten Zahl bezeichnet ist, also (B). In Abbildung 1.12 ist dies verzeichnet.



Von dieser Stadt aus werden wiederum die Entfernungen zu allen Nachbarn bestimmt. Da die Ameisen jedoch bis nach (B) bereits 34 km unterwegs waren, müssen diese addiert werden. Versuchen Sie es! Stoßen Sie auf ein Problem?

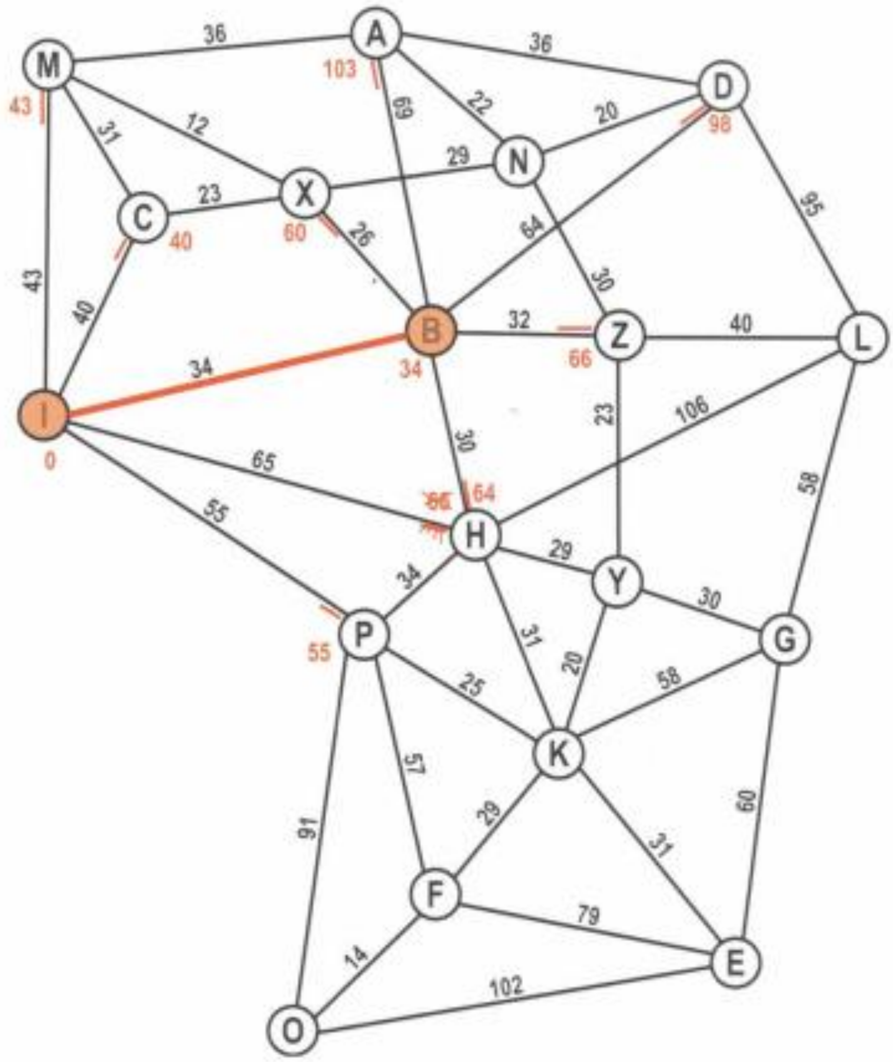
Genau! Für (X), (D) und (Z) können wir die Methodik einfach durchführen, aber was ist mit (H)? Hier steht bereits ein Wert!

Wie ist hier zu verfahren? Ziehen wir die Ameisen zurate: Der Weg über (B) nach (H) ist 64 km lang, der direkte Weg von (I) nach (H) 65 km. Der Ameisentrupp von (B) würde also VOR dem Trupp aus (I) ankommen und daher „gewinnen".

Daher gilt für das Dijkstra-Verfahren die folgende Regel für den Fall, dass an einer Nachbarstadt bereits eine Zahl steht:

- Wenn die Zahl, die neu hingeschrieben werden soll, kleiner ist, dann wird die alte durch die neue Zahl ersetzt und dementsprechend auch die Marke, woher die Ameisen kommen.
- Wenn die neue Zahl größer wäre, passiert gar nichts.

Ergebnis ist daher die Karte aus Abbildung 1.13.

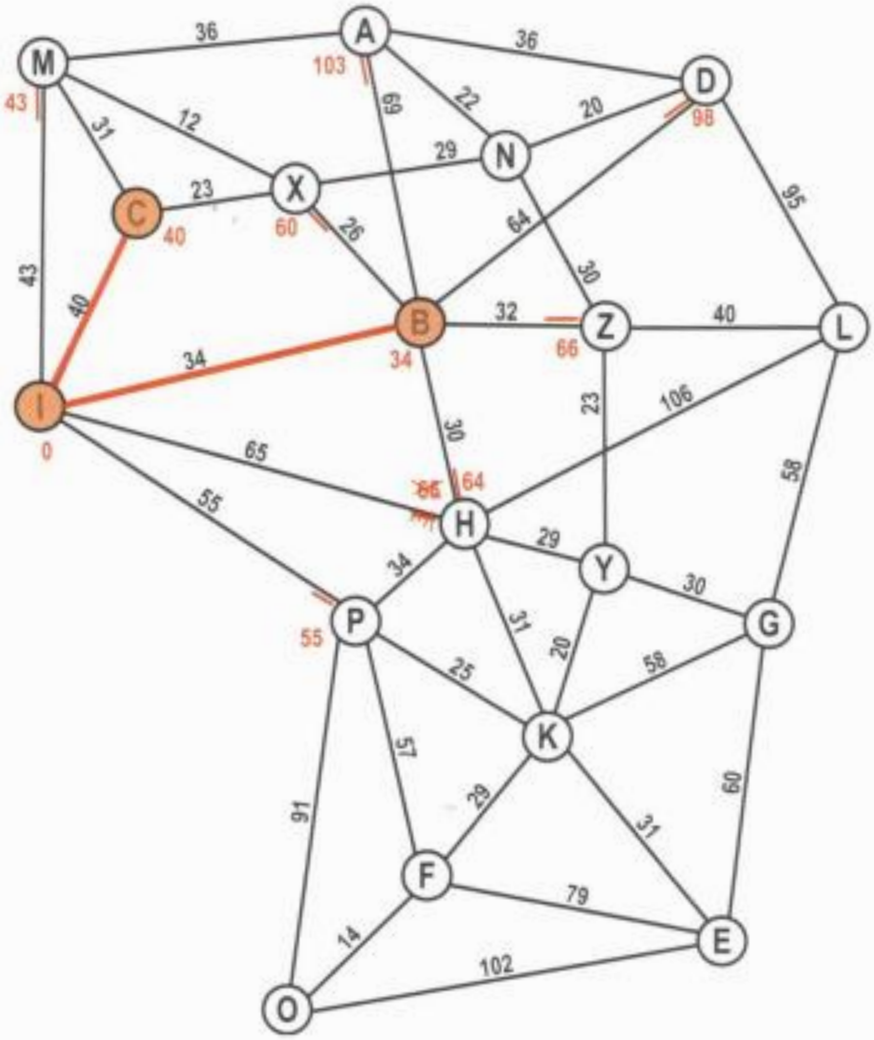


Wie geht es nun weiter?

Prinzipiell wie am Anfang: Aus allen mit Zahlen markierten Städten, die noch nicht von Ameisen besucht wurden (noch nicht rot gefärbt), wird die mit der kleinsten Zahl herausgesucht. Dort kommen die Ameisen als Nächstes an.

In diesem Fall ist das (C).

Von (C) aus werden wieder alle benachbarten Städte betrachtet. Nach (M) käme man in 71 km, nach (X) nach 63 km. Beides wird jedoch von der bereits vorhandenen Zahl unterboten, also passiert gar nichts weiter, die nächste nichtmarkierte Stadt mit der kleinsten Zahl wird gesucht. Das Ergebnis sehen Sie in Abbildung 1.14.



Bevor Sie nun weitermachen, versuchen Sie doch einmal, den Algorithmus zur Bestimmung kürzester Wege aufzuschreiben.

Algorithmen spielen in der Informatik eine sehr große Rolle: Ein Großteil der vorkommenden Aufgaben aus Wirtschaft und Wissenschaft lässt sich mit Algorithmen lösen, die schon vor längerer Zeit entwickelt wurden (dazu mehr am Ende des Kapitels).

Es gibt auch diverse formale Methoden, die Algorithmen zu Papier zu bringen. Eigentlich geht es jedoch darum, die Beschreibung so genau zu machen, dass jemand anderes den Algorithmus danach durchführen kann.

Behalten Sie das im Hinterkopf, wenn Sie jetzt versuchen, den beschriebenen Dijkstra-Algorithmus insgesamt zu formulieren.

Es gibt natürlich unzählige korrekte Lösungen. Ihre sollte im Kern mit Abbildung 1.15 übereinstimmen.

0. Markiere die **Startstadt** rot, weise ihr die **Kennzahl** 0 zu.
   Bezeichne diese als **aktuelle Stadt**.

1. Gehe aus von der **aktuellen Stadt** zu allen direkt erreichbaren **Nachbarstädten**.

   ... und führe das Folgende für jede **Nachbarstadt** durch:

   Errechne die **Summe** aus der Kennzahl an der **aktuellen Stadt** und der Länge der Strecke dorthin.

   - Ist die **Nachbarstadt** bereits rot markiert, mache nichts.
   - Hat die **Nachbarstadt** keine **Kennzahl**, weise ihr die **Summe** als **Kennzahl** zu und markiere die Strecke zur **aktuellen Stadt**.
   - Hat die **Nachbarstadt** eine **Kennzahl** kleiner der **Summe**, mache nichts.
   - Hat die **Nachbarstadt** eine **Kennzahl** größer der **Summe**, streiche die dortige **Kennzahl** sowie die Markierung. Weise ihr danach die **Summe** als neue **Kennzahl** zu. Markiere die Strecke zur **aktuellen Stadt**.

2. Betrachte alle Städte, die zwar eine **Kennzahl** haben, aber noch nicht rot markiert sind. Suche die Stadt mit der kleinsten **Kennzahl**.

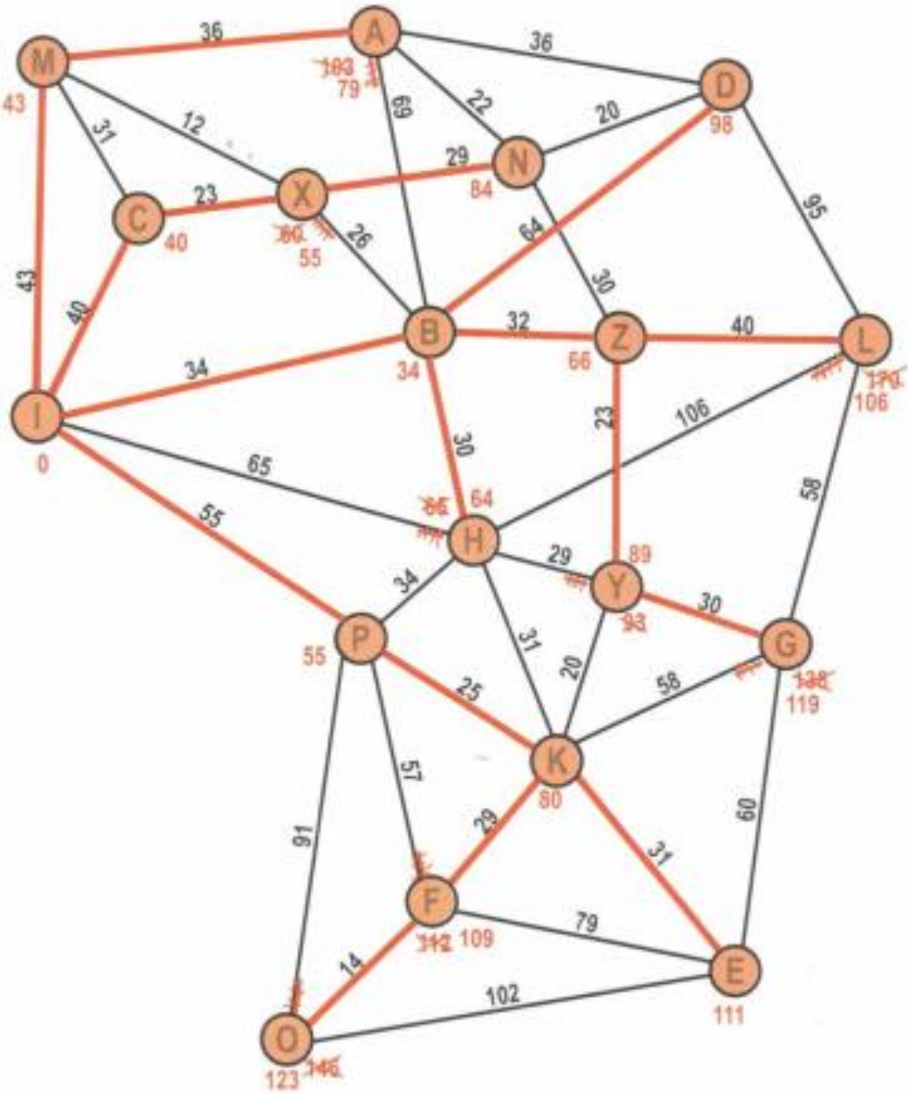
3. Bezeichne diese als **aktuelle Stadt**. Weisen mehrere Städte die kleinste **Kennzahl** auf, wähle eine beliebige davon als **aktuelle Stadt**.

4. Markiere die **aktuelle Stadt** rot,
   zeichne die dort markierte Strecke in rot ein.

5. Falls es noch Städte gibt, die nicht rot markiert sind,
   weiter bei (1.)

Führen Sie nun den Algorithmus für das gegebene Beispiel fort.

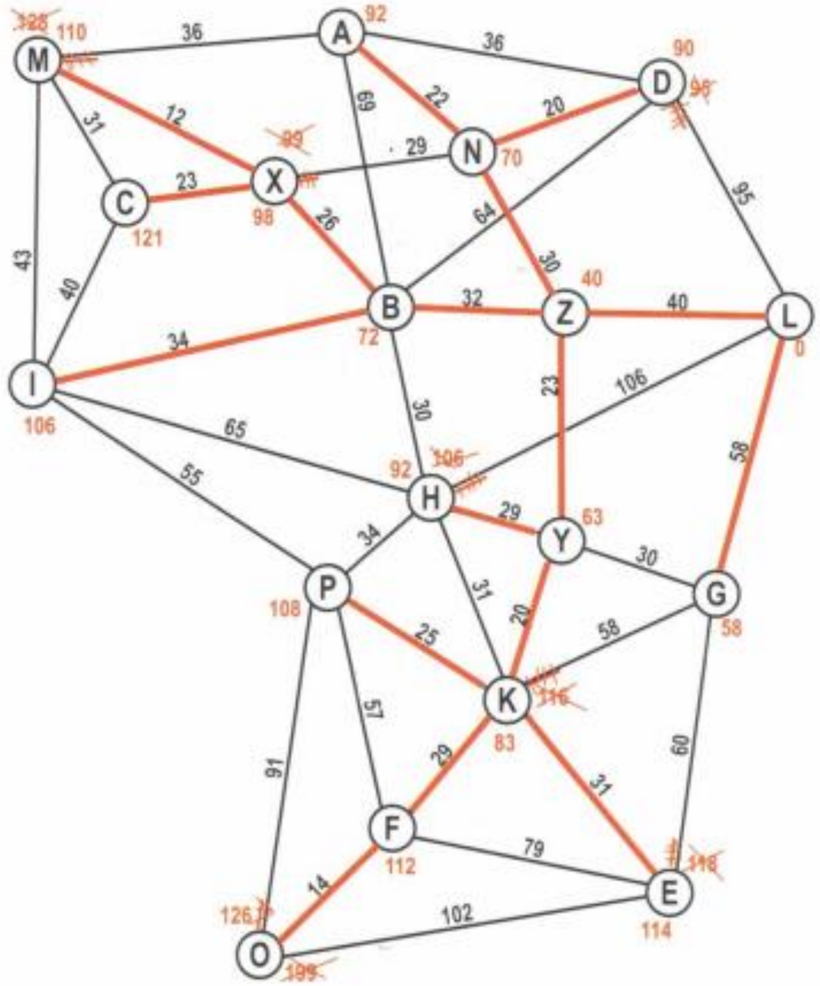
Das Ergebnis ist in Abbildung 1.16 dargestellt.



Erkennen Sie die Gemeinsamkeiten mit dem Ergebnis des Ameisen-Prinzips?

Üben Sie nun den neuen Algorithmus und bestimmen Sie den kürzesten Weg von Lupera nach Eindhofen, nach Giwelau und nach Morbach.
Zur Kontrolle: Die ermittelten Entfernungen sollten 58 km, 110 km und 114 km betragen.

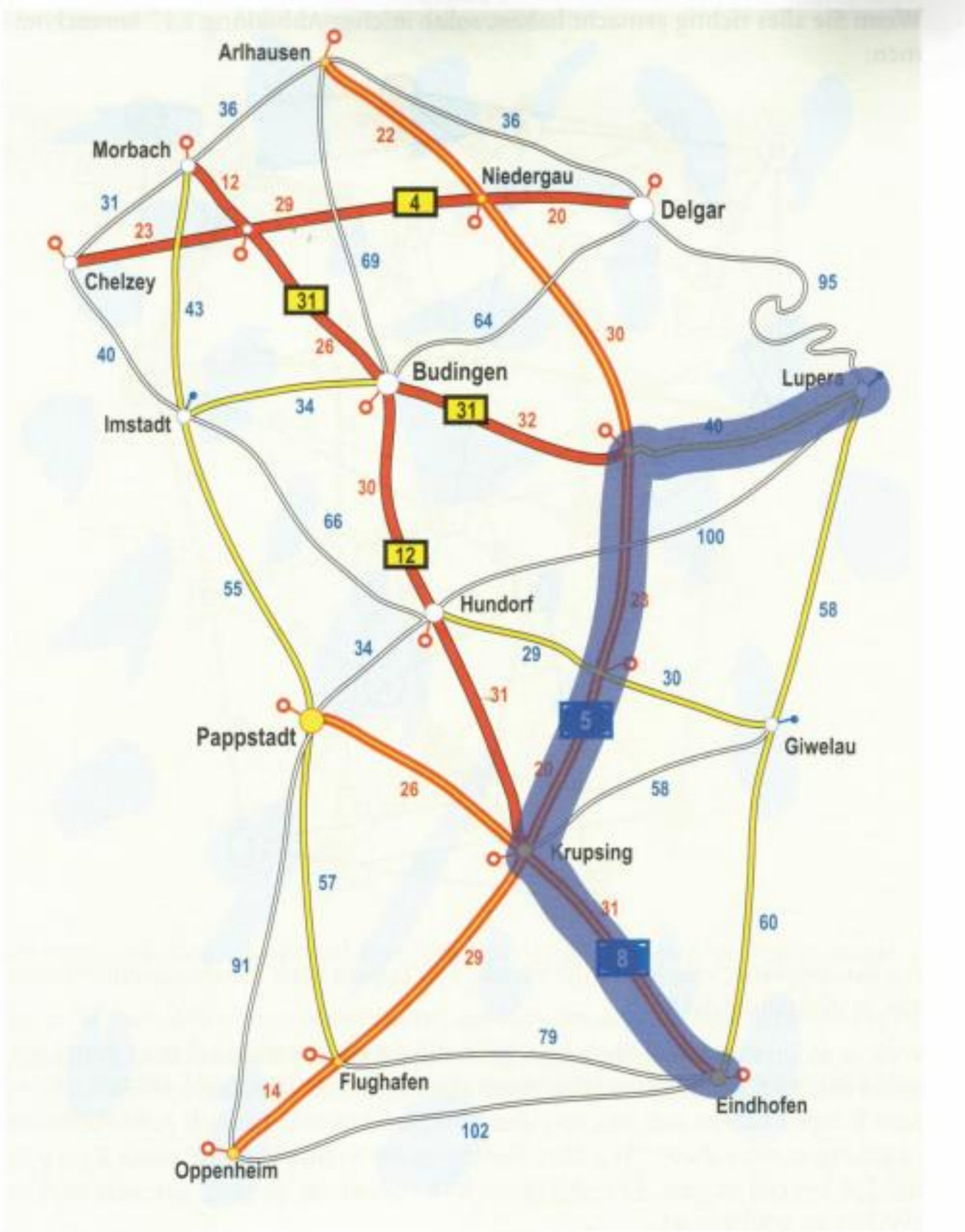
Wenn Sie alles richtig gemacht haben, sollte in etwa Abbildung 1.17 herauskommen.



Ein Routenplaner würde also die Strecke von Lupera nach Eindhofen einzeichnen wie in Abbildung 1.18.

Weil es ab und zu Verwechslungen gibt: Bitte denken Sie daran, dass das dargestellte rote Netz nur die kürzesten Wege von Lupera aus repräsentiert! Wenn Sie es zum Beispiel nutzen würden, um über rote Strecken von Imstadt zum Flughafen zu gelangen, wäre dieser Weg über Budingen, die Kreuze Z und Y sowie Krupsing mit 138 km viel zu weit. Aus Abbildung 1.16 können Sie ablesen, dass dies auch in 109 km zu schaffen ist.

Dijkstra bestimmt immer die kürzesten Wege von einem Startpunkt zu vielen anderen Punkten!



Herzlichen Glückwunsch – Sie wissen nun, wie so ein Routenplaner funktioniert!

Vorlage zum Benutzen