

Bundeswettbewerb Informatik: Aufgabe 3

Lösungsidee

Das Auftragsbuch ist ein Tourenplan.

Aus dem Tourenplan werden andere Pläne erstellt, aus denen man letztlich die Anzahl der anzumietenden Transporter ablesen kann:

Für alle Standorte und Wochentage wird die Anzahl der geplanten Abfahrten und die Anzahl der geplanten Ankünfte von Transportern berechnet.

$$Ab(\text{Tag}, A) = \text{Touren}(\text{Tag}, A \rightarrow B) + \text{Touren}(\text{Tag}, A \rightarrow C)$$

...

$$An(\text{Tag}, A) = \text{Touren}(\text{Tag}, B \rightarrow A) + \text{Touren}(\text{Tag}, C \rightarrow A)$$

...

Am Montag stehen noch keine Transporter zur Verfügung. Die zur Verfügung stehenden Transporter können mit folgender Formel berechnet werden:

$$\text{Transporter}(\text{Montag}, \text{Standort}) = 0$$

$$\begin{aligned} \text{Transporter}(\text{Tag}, \text{Standort}) &= \text{Transporter}(\text{Tag} - 1, \text{Standort}) + An(\text{Tag} - 1, \text{Standort}) \\ &\quad - Ab(\text{Tag} - 1, \text{Standort}) + \text{Bedarf}(\text{Tag} - 1, \text{Standort}) \end{aligned}$$

Die Anzahl der anzumietenden Transporter erhält man, indem man die Anzahl der zur Verfügung stehenden Transporter von der Anzahl der abfahrenden Transporter subtrahiert. Negative Werte würden dabei einen Überschuss an Transportern bedeuten und werden deshalb nicht berücksichtigt:

$$\text{Mieten}(\text{Tag}, \text{Standort}) = \text{Max}(Ab(\text{Tag}, \text{Standort}) - \text{Transporter}(\text{Tag}, \text{Standort}), 0)$$

Die *Max*-Funktion gibt den jeweils größeren Wert der beiden Argumente zurück.

Von nun an muss unterschieden werden, ob Leerfahrten gemacht werden dürfen oder nicht:

Ohne Leerfahrten

Die Anzahl der anzumietenden Transporter werden für jeden Standort aufsummiert:

$$\text{Mieten}_{\text{insgesamt}}(\text{Standort}) = \text{Mieten}(\text{Montag}, \text{Standort}) + \dots + \text{Mieten}(\text{Samstag}, \text{Standort})$$

Mit Leerfahrten

Zusätzlich werden die überschüssigen Transporter berechnet. Negative Werte würden dabei einen Mangel an Transportern bedeuten und werden deshalb nicht berücksichtigt:

$$\text{Überschuss}(\text{Tag}, \text{Standort}) = \text{Max}(\text{Transporter}(\text{Tag}, \text{Standort}) - Ab(\text{Tag}, \text{Standort}), 0)$$

Nun hat man alle Informationen, um eine eventuelle Leerfahrt zu erkennen. Eine Leerfahrt wird immer dann gemacht, wenn Transporter am „nächsten Tag“ angemietet werden müssten, es aber am „heutigen Tag“ einen Überschuss an Transportern gibt.

Allgemein kann man die Anzahl der Leerfahrten mit Hilfe folgender Formel berechnen:

$$\text{Leerfahrten}(\text{Tag}, \text{Von}, \text{Nach}) = \text{Min}(\text{Überschuss}(\text{Tag}, \text{Von}), \text{Mieten}(\text{Tag} - 1, \text{Nach}))$$

Die Variablen *Von* und *Nach* stehen für Standorte. Die Anzahl der Leerfahrten vom Standort A nach Standort B kann also mit *Leerfahrten* (*Tag*, *A*, *B*) berechnet werden.

Die Funktion *Min* gibt den jeweils kleineren Wert der beiden Argumente zurück. Die *Min*-Funktion wird benötigt, um zu erkennen, ob eine Leerfahrt gemacht werden muss: Gibt es keinen Überschuss an Transportern oder müssen keine Transporter angemietet werden, dann müssen auch keine Leerfahrten gemacht werden. Gleichzeitig bewirkt die *Min*-Funktion, dass nur so viele Leerfahrten gemacht werden, wie es überschüssige Transporter gibt. Es werden auch nie mehr Leerfahrten gemacht, als am nächsten Tag Transporter benötigt werden.

Beim ersten Argument der *Min*-Funktion müssen die bereits berechneten Leerfahrten, die als Abfahrtsort den Standort *Von* haben abgezogen werden. Beim zweiten Argument der *Min*-Funktion müssen die bereits berechneten Leerfahrten, die als Zielort den Standort *Nach* haben abgezogen werden. Um z. B. die Anzahl der Leerfahrten von Standort B zu Standort C zu berechnen muss beim ersten Argument der *Min*-Funktion die Anzahl der Leerfahrten von B nach A und beim zweiten Argument die Anzahl der Leerfahrten von A nach C abgezogen werden.

Sobald eine Leerfahrt „erkannt“ wurde, wird diese im Tourenplan eingetragen. Dies hat zur Folge, dass alle Werte (*Ab*, *An*, *Transporter*, *Mieten*, *Überschuss*, *Leerfahrten*) neu berechnet werden müssen. Dies wird solange wiederholt, bis keine Leerfahrten mehr auftreten.

Kann die Anzahl der anzumietenden Transporter trotz Leerfahrten nicht ausgeglichen werden, müssen Transporter angemietet werden.

Zum Schluss kann die Anzahl der anzumietenden Transporter mit der selben Formel berechnet werden, wie sie bei *Ohne Leerfahrten* zu sehen ist.

Beispiel

Die Lösungsidee wird an einem Beispiel veranschaulicht:

Das Auftragsbuch bzw. der Tourenplan:

| Wochentag | A nach B | A nach C | B nach A | B nach C | C nach A | C nach B |
|------------|----------|----------|----------|----------|----------|----------|
| Montag | 1 | 2 | 3 | 4 | 5 | 6 |
| Dienstag | 2 | 3 | 4 | 5 | 6 | 1 |
| Mittwoch | 3 | 4 | 5 | 6 | 7 | 8 |
| Donnerstag | 1 | 2 | 3 | 4 | 5 | 6 |
| Freitag | 1 | 2 | 3 | 4 | 5 | 6 |
| Samstag | 4 | 3 | 5 | 6 | 1 | 2 |

Die Anzahl der abfahrenden und ankommenden Transporter für jeden Tag:

| Wochentag | ab A | an A | ab B | an B | ab C | an C |
|------------|------|------|------|------|------|------|
| Montag | 3 | 8 | 7 | 7 | 11 | 6 |
| Dienstag | 5 | 10 | 9 | 3 | 7 | 8 |
| Mittwoch | 7 | 12 | 11 | 11 | 15 | 10 |
| Donnerstag | 3 | 8 | 7 | 7 | 11 | 6 |
| Freitag | 3 | 8 | 7 | 7 | 11 | 6 |
| Samstag | 7 | 6 | 11 | 6 | 3 | 9 |

Für ein Wochentag wird zunächst die Anzahl der zur Verfügung stehenden Transporter berechnet und danach die Anzahl der anzumietenden Transporter. Die Berechnung der anzumietenden Transporter benötigt nämlich die Anzahl der zur Verfügung stehenden Transporter vom Vortag.

Die Anzahl der zur Verfügung stehenden Transporter:

| Wochentag | Standort A | Standort B | Standort C |
|------------|------------|------------|------------|
| Montag | 0 | 0 | 0 |
| Dienstag | 8 | 7 | 6 |
| Mittwoch | 13 | 3 | 8 |
| Donnerstag | 18 | 11 | 10 |
| Freitag | 23 | 11 | 6 |
| Samstag | 28 | 11 | 6 |
| Sonntag | 27 | 6 | 12 |

Die Anzahl der anzumietenden Transporter:

| Wochentag | Standort A | Standort B | Standort C |
|------------|------------|------------|------------|
| Montag | 3 | 7 | 11 |
| Dienstag | 0 | 2 | 1 |
| Mittwoch | 0 | 8 | 7 |
| Donnerstag | 0 | 0 | 1 |
| Freitag | 0 | 0 | 5 |
| Samstag | 0 | 0 | 0 |

Summiert man die Anzahl der anzumietenden Transporter spaltenweise auf, so erhält man die zu Beginn der Woche anzumietenden Transporter, wenn keine Leerfahrten gemacht werden dürfen.

Die überschüssigen Transporter:

| Wochentag | Standort A | Standort B | Standort C |
|------------|------------|------------|------------|
| Montag | 0 | 0 | 0 |
| Dienstag | 3 | 0 | 0 |
| Mittwoch | 6 | 0 | 0 |
| Donnerstag | 15 | 4 | 0 |
| Freitag | 20 | 4 | 0 |
| Samstag | 21 | 0 | 3 |

Am Montag muss angemietet werden, da noch keine Transporter zur Verfügung stehen.

Die erste Leerfahrt tritt am Dienstag auf, denn dort gibt es einen Überschuss von 3 Transportern und am Mittwoch würden 8 Transporter für Standort B angemietet werden. Die *Min*-Funktion liefert den Wert 3, denn es sind nur 3 Transporter als Überschuss vorhanden. Die Leerfahrten werden im Tourenplan hinzugefügt, indem die Anzahl der von A nach B fahrenden Transporter um 3 erhöht wird. Nun müssen alle Tabellen neu berechnet und wieder nach Leerfahrten gesucht werden. Der Überschuss vom Dienstag wurde komplett verbraucht. Dennoch müssen Transporter für alle drei Standorte angemietet werden. Dann beginnt man wieder mit der Suche von Leerfahrten. Dieses Verfahren wird solange wiederholt, bis keine Leerfahrten mehr gemacht werden können.

Programm-Dokumentation

Die Lösungsidee ist von mir in der Programmiersprache C umgesetzt worden.

Das Auftragsbuch ist ein zweidimensionales Array. Es wird durch die Funktion **auftragsbuch_erstellen** erstellt. Die Funktion muss mit einem Dateinamen als Parameter aufgerufen werden. Die Funktion liest die Daten für das Auftragsbuch aus der Datei und speichert die Daten in dem zweidimensionalen Array.

Den Funktionen **ohne_leerfahrten** und **mit_leerfahrten** bekommen das Auftragsbuch als Parameter übergeben. Die beiden Funktionen berechnen die zum Anfang der Woche anzumietenden Transporter für die jeweiligen Standorte. Die Funktion **ohne_leerfahrten** gibt die Ergebnisse für die erste Teilaufgabe und die Funktion **mit_leerfahrten** für die zweite Teilaufgabe auf dem Bildschirm aus.

In den Funktionen **ohne_leerfahrten**, **mit_leerfahrten** werden Arrays erstellt, die die Ergebnisse der Funktionen *An*, *Ab*, *Transporter*, *Mieten*, *Überschuss*, *Leerfahrten* und *Mieten_{insgesamt}* speichern. Wie in der Lösungsidee beschrieben, werden die Daten des Auftragsbuches verarbeitet und die Ergebnisse in den Arrays gespeichert. Die Ergebnisse werden dann auf dem Bildschirm ausgegeben.

Die Funktion **berechne_tabellen** ist für die Initialisierung und Aktualisierung der Werte in den Arrays zuständig. Die Funktion beinhaltet die Formeln: *An*, *Ab*, *Transporter*, *Mieten* und *Überschuss*. Die Funktion verarbeitet also die Daten aus dem Auftragsbuch. Nach jeder erkannten Leerfahrt wird diese Funktion aufgerufen. Dann wird wieder nach Leerfahrten gesucht. Es wird immer beginnend bei dem Montag nach Leerfahrten gesucht. Dieser Vorgang wiederholt sich solange, bis keine Leerfahrten mehr gemacht werden können. Zum Schluss enthalten die Arrays die Ergebnisse, die dann auf dem Bildschirm ausgegeben werden.

Die Funktion **mit_leerfahrten** ändert das Auftragsbuch zu dem Tourenplan, der die Leerfahrten enthält. Das veränderte Auftragsbuch wird dann mit der Funktion **print_auftragsbuch** ausgegeben.

Programm-Ablaufprotokoll

Das Programm wird mit dem Dateinamen eines Auftragsbuches aufgerufen. Das Programm berechnet die Ergebnisse und gibt dann folgendes auf dem Bildschirm aus:

```
$ ./logistisch logistisch1.txt
```

Tourenplan:

=====

| | A->B | A->C | B->A | B->C | C->A | C->B |
|-------------|------|------|------|------|------|------|
| Montag: | 1 | 2 | 3 | 4 | 5 | 6 |
| Dienstag: | 2 | 3 | 4 | 5 | 6 | 1 |
| Mittwoch: | 3 | 4 | 5 | 6 | 7 | 8 |
| Donnerstag: | 1 | 2 | 3 | 4 | 5 | 6 |
| Freitag: | 1 | 2 | 3 | 4 | 5 | 6 |
| Samstag: | 4 | 3 | 5 | 6 | 1 | 2 |
| Sonntag: | 0 | 0 | 0 | 0 | 0 | 0 |

ohne Leerfahrten:

=====

```
Montag:      3 Transporter fuer Standort A benoetigt
Montag:      7 Transporter fuer Standort B benoetigt
Montag:     11 Transporter fuer Standort C benoetigt
Dienstag:     2 Transporter fuer Standort B benoetigt
Dienstag:     1 Transporter fuer Standort C benoetigt
Mittwoch:     8 Transporter fuer Standort B benoetigt
Mittwoch:     7 Transporter fuer Standort C benoetigt
Donnerstag:    1 Transporter fuer Standort C benoetigt
Freitag:      5 Transporter fuer Standort C benoetigt
```

Am Ende der Woche:

```
  27 Transporter am Standort A
   6 Transporter am Standort B
  12 Transporter am Standort C
```

zu Beginn der Woche 45 Transporter anmieten:

```
  3 Transporter fuer Standort A
 17 Transporter fuer Standort B
 25 Transporter fuer Standort C
```

mit Leerfahrten:

=====

```
Montag:      3 Transporter fuer Standort A benoetigt
Montag:      7 Transporter fuer Standort B benoetigt
Montag:     11 Transporter fuer Standort C benoetigt
Dienstag:     2 Transporter fuer Standort B benoetigt
Dienstag:     1 Transporter fuer Standort C benoetigt
Dienstag:     3 Leerfahrten von A nach B
Mittwoch:     5 Transporter fuer Standort B benoetigt
Mittwoch:     7 Transporter fuer Standort C benoetigt
```

Mittwoch: 1 Leerfahrten von A nach C
 Donnerstag: 5 Leerfahrten von A nach C

Am Ende der Woche:

18 Transporter am Standort A
 6 Transporter am Standort B
 12 Transporter am Standort C

zu Beginn der Woche 36 Transporter anmieten:

3 Transporter fuer Standort A
 14 Transporter fuer Standort B
 19 Transporter fuer Standort C

Tourenplan (inklusive Leerfahrten):

| | A->B | A->C | B->A | B->C | C->A | C->B |
|-------------|------|------|------|------|------|------|
| Montag: | 1 | 2 | 3 | 4 | 5 | 6 |
| Dienstag: | 5 | 3 | 4 | 5 | 6 | 1 |
| Mittwoch: | 3 | 5 | 5 | 6 | 7 | 8 |
| Donnerstag: | 1 | 7 | 3 | 4 | 5 | 6 |
| Freitag: | 1 | 2 | 3 | 4 | 5 | 6 |
| Samstag: | 4 | 3 | 5 | 6 | 1 | 2 |
| Sonntag: | 0 | 0 | 0 | 0 | 0 | 0 |