
Aufgabe D1: Rundreise

Ein Handlungsreisender muss eine festgelegte Anzahl von Orten besuchen (dabei sei `anzOrte` ein konstanter Wert größer 1). Er möchte dafür eine Rundreise planen, während der jeder Ort genau einmal besucht wird und die wieder in dem Ort endet, in der sie begonnen hat. Aus Sparsamkeits- und Umweltschutzgründen sollte die Reise möglichst kurz sein.

Schreiben Sie hierfür eine Funktion

```
func rundreise(entfernung [anzOrte][anzOrte]int) ([anzOrte + 1]int, int)
    { ... }
```

welche als Eingabe eine Entfernungsmatrix erhält, daraus eine Rundreise ermittelt und abschließend diese Route sowie die insgesamt zu fahrenden Kilometer zurückgibt. Die Route soll dabei in Form eines Arrays zurückgegeben werden, welches die Ortsnummern in der Reihenfolge der Rundreise enthält. Die Nummerierung der Orte soll dabei mit 1 beginnen (d.h. der erste Ort in der Matrix mit Array-Index 0 hat die Ortsnummer 1, usw.).

Zur Berechnung der Rundreise soll folgendes Verfahren benutzt werden: Ausgehend vom Ort 1 wird jeweils zum *nächstgelegenen* Ort gefahren, bis alle Orte besucht sind. Zum Schluss muss dann wieder zum Ort 1 zurückgekehrt werden. Die Entfernungen zwischen den Orten sind in der übergebenen Matrix abgespeichert. Die Information, ob ein Ort schon besucht ist, soll in einem booleschen Array abgelegt werden.

Für beispielsweise `anzOrte = 3` und die Entfernungsmatrix

```
1 2 3
1 [0 7 4]
2 [7 0 6]
3 [4 6 0]
```

ergeben sich die folgenden Resultate:

```
Route der Rundreise: [1,3,2,1]
Gesamtstrecke:      17
```

Sie können davon ausgehen, dass die Länge der Rundreise und die einzelnen Entfernungen zwischen den Orten natürliche Zahlen kleiner als `math.MaxInt` sind. Alle Entfernungen zwischen den Orten sind unterschiedlich.

Ressourcen

Im Ordner dieser Aufgabe finden Sie eine Datei `rundreise.go`, die Sie entsprechend der Aufgabenstellung abändern sollen.

Weiterhin stellen wir in der Datei `main.go` eine `main`-Funktion zur Verfügung, damit Sie Ihre Funktion in einem beispielhaften Kontext kompilieren und ausführen können. Nachdem Sie in den Aufgabenordner gewechselt sind, geben Sie dazu folgenden Befehl ein:

```
go run .
```

Die Datei `rundreise_test.go` stellt Tests bereit, die Sie mit folgendem Befehl durchführen können:

```
go test
```

Im Unterordner `ML` finden Sie einen Lösungsvorschlag.