

# Aufgabe 4: Urlaubsfahrt

Team-ID: 00587

Team-Name: Doge.NET

Bearbeiter dieser Aufgabe:  
Johannes von Stoephasius  
Nikolas Kilian

22. November 2019

## Inhaltsverzeichnis

## 1 Lösungsidee

### 1.1 Definitionen

**Weg** Ein "Weg" ist eine Menge von zu besuchenden Tankstellen .

**optimaler Weg** Der "beste Weg" oder "optimale Weg", ist der Weg mit den wenigsten Stopps, und geringsten Preis um vom Ausgangspunkt zu einer gegebenen Tankstelle zu kommen. Hierbei wird wie in der Aufgabenstellung eine minimal Stoppzahl priorisiert.

### 1.2 Kernidee

«««< HEAD Das finden des optimalen Wegs kann in zwei Teile getrennt werden; das finden aller Wege mit minimaler Stoppzahl, und das ermitteln der billigsten dieser.

Der Algorithmus funktioniert, indem er nach und nach den besten Weg zu jeder Tankstellen findet, wobei er zuerst den besten Weg zu den Tankstellen am nächsten zum Startpunkt findet, bishin zu zuletzt der letzten. ===== Der Algorithmus basiert auf der Idee durch die Lösung von Teilproblemen die insgesamt beste Lösung zu finden. Das heißt, dass zur Findung des besten Weges zu einem Punkt ein neuer Weg aus dem besten Weg zu den Punkten davor gebildet wird. Zum Finden des besten Weges zu einm Punkt wird der Weg von den Wegen zu den Punkten davor gewählt, der am wenigsten Stopps braucht und den geringsten Preis hat. Durch den iterativen Aufbau von den besten Wegen ist garantiert, dass am Ende der beste Weg zum Ziel gefunden wird.

### 1.3 Finden des besten Weges zum Ziel

»»»> fc60c9de820be25225ad8fbaa39b724c9abb3733

Zum finden des besten Weges zum Ziel wird nach und nach eine Zuordnung aufgebaut, die für jede Tankstelle den besten Weg zu ihr enthält.

Diese Zuordnung wird für eine beliebige Tankstelle ermittelt, indem zuerst für alle vorherigen Tankstellen eine Zuordnung ermittelt wird.

Sind die optimalen Wege zu allen Tankstellen vor einer gegebenen Tankstelle errechnet, kann die Ermittlung des optimalen Wegs zu dieser beginnen. Zuerst werden alle optimale Wege zu den vorherigen Tankstelle durchiteriert. Dabei wird zuerst überprüft von welchen dieser Tankstellen die aktuell betrachtete Tankstelle überhaupt erreicht werden kann.

Wenn nicht wird zur nächsten Tankstelle der Zuordnung übergegangen.

Wenn doch, so wird ein neuer Weg gebildet, der aus dem vorherigen Weg und der betrachtete Tankstelle besteht.

Von allen dieser neuen Wege werden nun die kürzesten ausgewählt, also alle Wege mit einer Stoppzahl gleich der minimalen Stoppzahl. Danach wird der Preis aller übrigen Wege ermittelt, und von denen wird der billigste ausgewählt. Dieser Weg ist dann der optimale Weg vom Start zur aktuell betrachteten Tankstelle. Dieser wird dann in die Zuordnung eingetragen.

Dieses Verfahren wird für alle Tankstellen iterativ durchgeführt, bis der optimale Weg zum Streckenende Teil der Zuordnung ist. Dieser ist dann der optimale Weg vom Start zum Streckenende.

### 1.4 Ermittlung des Preises für einen Weg

Für die Ermittlung des Preises für einen Weg werden vorerst die Tankstellen dem Preis nach aufsteigend geordnet. Zuerst wird von der preiswertesten Tankstelle aus eine Strecke definiert, die von der Tankstelle aus über die maximalen Tanklänge reicht, oder wenn das Streckenende in dieser enthalten ist, bis zum Ziel geht.

Danach wird für die nächst-preiswerteste Tankstelle auch eine solche Strecke definiert, es sei denn, es gibt Überschneidungen mit einer bereits eingetragenen Strecke. In diesem Fall wird die bereits eingetragene Strecke durch maximales Volltanken des halb-leeren Tanks an dieser Tankstelle erweitert.

Dieses Verfahren wird so lange wiederholt, bis nur noch eine Strecke existiert, die den gesamten Weg abdeckt. Dieses Verfahren liefert das ideale Ergebnis, da preislich aufsteigend immer die beste Teillösung gefunden wird.

## 2 Umsetzung

### 2.1 Codestruktur

- Urlaubsfahrt.cs
  - enthält die statische Klasse `Urlaubsfahrt`, die die statische Methode `GetTrack` enthält, die den Hauptalgorithmus ausführt
  - `GetTrack` erhält als Parameter:
    - \* die Strecke, die mit dem bereits im Tank vorhandenen Benzin zurückgelegt werden kann
    - \* die mit einer maximalen Tankfüllung zurückgelegt werden kann
    - \* eine Liste aller Tankstellen
- GasStation.cs
  - enthält die Position und den Benzinpreis einer Tankstelle
- Track.cs
  - berechnet für einen Weg den besten Preis, unabhängig von den benötigten Stopps
  - enthält eine Liste von allen Tankstellen des Weges
  - ein Track kann folgendermaßen gebildet werden:
    - \* aus einer Tankstelle
    - \* aus einem Track und einer Tankstelle
    - \* aus nichts; das statische Feld `EmptyTrack` repräsentiert einen komplett leeren Weg
  - die Methode `GetPriceTo` führt den Nebenalgorithmus aus, und berechnet entweder bis zu einem arbiträren Punkt, oder zu einer Tankstelle den Preis aus
- Extensions.cs
  - enthält Methoden, die generische Klassen oder Interfaces erweitern
    - \* `AllMins<TSource>` gibt von einem `IEnumerable` von `TSource` die Elemente zurück, die bei einer Funktion von `TSource` zu `Comparable<T>` gleichwertig zum Minimum sind

#### 2.1.1 Implementierung

```
1 GetTrack(float MitDemStartBenzinZurücklegbarerWeg, float MitVollemTankZurücklegbarerWeg,
2         Liste<Tankstelle> AlleTankstellen)
3 {
4     Liste<Weg> WegeZuAllenTankstellen = Weg.LeererWeg;
5     FürJede(Tankstelle Station in AlleTankstellen)
6     {
7         WegeZuAllenTankstellen.EntferneAlleFürDieGilt((der Abstand der letzten
8         Tankstelle zu Station > MitVollemTankZurücklegbarerWeg);
9     }
10 }
```

## 3 Beispiele

Genügend Beispiele einbinden! Die Beispiele von der BwInf-Webseite sollten hier diskutiert werden, aber auch eigene Beispiele sind sehr gut – besonders wenn sie Spezialfälle abdecken. Aber bitte nicht 30 Seiten Programmausgabe hier einfügen!

## 4 Quellcode

Unwichtige Teile des Programms sollen hier nicht abgedruckt werden. Dieser Teil sollte nicht mehr als 2–3 Seiten umfassen, maximal 10.