

Aufgabe 4: Urlaubsfahrt

Team-ID: 00587

Team-Name: Doge.NET

Bearbeiter dieser Aufgabe:
Johannes von Stoephasius

19. November 2019

Inhaltsverzeichnis

1	Lösungsidee	1
2	Umsetzung	2
2.1	Codestruktur	2
2.1.1	Implementierung	2
3	Beispiele	2
4	Quellcode	2

1 Lösungsidee

Im Folgenden wird "Weg" als Alias für eine Menge von Tankstellen verwendet, von der man den günstigsten Preis um zur letzten Tankstelle zu kommen, abrufen kann.

Der Algorithmus ist in 2 Teile geteilt, wobei der Nebenalgorithmus den günstigsten Preis eines Weges berechnet und der Hauptalgorithmus den insgesamt besten Weg zum Ziel findet. Der Hauptalgorithmus enthält eine Liste, die für jede Tankstelle den besten Weg (die wenigsten Stopps und der geringste Preis) zu ihr enthält. Jeder dieser Wege wird gebildet, indem für jede Tankstelle davor der beste Weg gefunden wird. Aus all diesen bereits berechneten Wegen wird jeweils ein neuer gebildet, indem die aktuell betrachtete Tankstelle angehängt wird, von denen zuerst die mit den wenigsten Stopps ausgewählt werden und dann die, die am günstigsten sind. Die Günstigsten werden mit Hilfe vom Nebenalgorithmus gefunden. Diese Wege werden dann der Liste hinzugefügt und das Verfahren wird für die nächste Tankstelle wiederholt, bis das Ziel erreicht wird.

Der Nebenalgorithmus berechnet den besten Preis für einen Weg. Dafür werden die Tankstellen dem Preis entlang aufsteigend geordnet. Zuerst wird von der preiswertesten Tankstelle aus eine Range definiert, die von der Tankstelle aus bis zur maximalen Tanklänge, oder wenn die Gesamt-Streckenlänge kürzer ist, bis zum Streckenende geht. Danach wird für die zweit-preiswerteste Tankstelle auch eine solche Range definiert, es sei denn, es gibt Überschneidungen mit einer bereits gefundenen Range. In diesem Fall wird die bereits gefundene Range erweitert. Dieses Verfahren wird so lange wiederholt, bis nur noch eine Range 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 der Hauptalgorithmus ist
 - 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:
 - * nur aus einer Tankstelle
 - * aus einem Track und einer Tankstelle
 - * weiter existiert ein Empty-Track, der keine Stationen beinhaltet
 - die Methode GetPriceTo ist der Nebenalgorithmus und berechnet entweder bis zu einem Punkt oder zu einer Tankstelle den Preis
 - Extensions.cs
 - * enthält Methoden, die generische Klassen oder Interfaces erweitern
 - AllMins gibt von einem IEnumerable von TSource die Elemente zurück, die bei einer Funktion von TSource zu einem IComparable minimal sind

2.1.1 Implementierung

```

1 GetTrack(float MitDemStartBenzinZur[U+FFFD]cklegbarerWeg, float MitVollemTankZur[U+FFFD]cklegbarerWeg, Liste<Tankst
  {
3   Liste<Weg> WegeZuAllenTankstellen = Weg.LeererWeg;
   F(U+FFFD)cklegbarerWeg)
5   {
       WegeZuAllenTankstellen.EntferneAlleF(U+FFFD)rDieGilt(der Abstand der letzten Tankstelle zu Station > MitVo
7   }
  }

```

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.