Dokumentation

Aufgabe 1: „Faires Füllen“

**Lösungsidee:** Ausgehend vom Startzustand sollen durch die verschiedenen Umfüllaktionen alle möglichen Zustände erreicht und gespeichert werden. Ist der Endzustand - eine faire Verteilung – in dieser Zustandssammmlung, so gibt es eine Lösung. Die Information, von welchem Zustand man durch Umfüllen in welche Zustände kommt, sollen in einen Graphen eingetragen werden.   
Schließlich soll das Programm die kürzeste Verbindung in diesem Graphen suchen.  
   
Das Programm soll also folgendes tun:

1. Einen Endzustand finden
2. Anhand der Anzahl der Gefäße alle möglichen Umfüllaktionen bestimmen
3. Mit dem Startzustand beginnend alle möglichen Zustände finden
4. Überprüfen, ob der Endzustand gefunden worden ist
5. Wenn es eine Lösung gibt, aus den Zuständen und ihren Nachbarzuständen einen Graphen erstellen
6. In diesem Graphen nach dem kürzesten Weg suchen
7. Das Ergebnis benutzerfreundlich ausgeben

Das Finden des Endzustandes ist simpel: Zwar gibt es oftmals verschiedene Möglichkeiten, die Gefäße fair zu füllen, jedoch lässt sich aus diesen Möglichkeiten durch Umfüllen immer folgende Verteilung erreichen: Ausgehend von einem beliebigem Gefäß der Person jedes Gefäß maximal füllen, bis diese Person genug (d.h. die Hälfte des Gesamtvolumens) erhalten hat.

**Umsetzung:**

Die Umsetzung der Idee erfolgte in Delphi.

Der Hauptteil des Programms wird durch die 2 Prozeduren „btMain.Click“ und „FindBestSolution“ ausgeführt. Erstere wird durch den Benutzer gestartet, lässt I in einer anderen Prozedur („FindProperEndzustand“) erledigen und bearbeitet II, III und IV. Danach ruft es die Prozedur „FindBestSolution“ auf, die die letzten Schritte erledigt.

Besondere Bedeutung hat die Funktion „DoUmfuellaktion“, die den Zustand angibt, der sich ergibt, wenn man einen ausgehend von einem Zustand eine Umfüllaktion ausführt.

**Quellcode:**

Der Übersichtlichkeit zuliebe, findet sich der Quellcode nicht in dieser Dokumentation. Nutzen Sie bitte die beiliegende Datei „TestUmfuellenMehrAls3.pas“. Es empfiehlt sich, entweder die Delphi Umgebung oder Notepad++ (Spracheinstellung Pascal) zu nutzen.

Die Lösungen des Programms für die 3 Beispielaufgaben.

**Beispiel 3**

Startzustand: Zustand: [0,0,0,20,]

Umfüllen: Von Gefäß 4 (50 Liter) ins Gefäß 1 (6 Liter)

Zustand: [6,0,0,14,]

Umfüllen: Von Gefäß 1 (6 Liter) ins Gefäß 3 (13 Liter)

Zustand: [0,0,6,14,]

Umfüllen: Von Gefäß 4 (50 Liter) ins Gefäß 1 (6 Liter)

Zustand: [6,0,6,8,]

Umfüllen: Von Gefäß 1 (6 Liter) ins Gefäß 3 (13 Liter)

Zustand: [0,0,12,8,]

Umfüllen: Von Gefäß 4 (50 Liter) ins Gefäß 1 (6 Liter)

Zustand: [6,0,12,2,]

Umfüllen: Von Gefäß 1 (6 Liter) ins Gefäß 3 (13 Liter)

Zustand: [5,0,13,2,]

Umfüllen: Von Gefäß 3 (13 Liter) ins Gefäß 4 (50 Liter)

Zustand: [5,0,0,15,]

Umfüllen: Von Gefäß 1 (6 Liter) ins Gefäß 3 (13 Liter)

Zustand: [0,0,5,15,]

Umfüllen: Von Gefäß 4 (50 Liter) ins Gefäß 1 (6 Liter)

Zustand: [6,0,5,9,]

Umfüllen: Von Gefäß 1 (6 Liter) ins Gefäß 3 (13 Liter)

Zustand: [0,0,11,9,]

Umfüllen: Von Gefäß 4 (50 Liter) ins Gefäß 1 (6 Liter)

Zustand: [6,0,11,3,]

Umfüllen: Von Gefäß 1 (6 Liter) ins Gefäß 3 (13 Liter)

Zustand: [4,0,13,3,]

Umfüllen: Von Gefäß 1 (6 Liter) ins Gefäß 2 (26 Liter)

Zustand: [0,4,13,3,]

Umfüllen: Von Gefäß 3 (13 Liter) ins Gefäß 1 (6 Liter)

Zustand: [6,4,7,3,]

Umfüllen: Von Gefäß 4 (50 Liter) ins Gefäß 3 (13 Liter)

Zustand: [6,4,10,0,]

Ziel erreicht

**Beispiel 2**

Es gibt leider KEINE Lösung

**Beispiel 1**

Startzustand: Zustand: [8,0,0,]

Umfüllen: Von Gefäß 1 (8 Liter) ins Gefäß 2 (5 Liter)

Zustand: [3,5,0,]

Umfüllen: Von Gefäß 2 (5 Liter) ins Gefäß 3 (3 Liter)

Zustand: [3,2,3,]

Umfüllen: Von Gefäß 3 (3 Liter) ins Gefäß 1 (8 Liter)

Zustand: [6,2,0,]

Umfüllen: Von Gefäß 2 (5 Liter) ins Gefäß 3 (3 Liter)

Zustand: [6,0,2,]

Umfüllen: Von Gefäß 1 (8 Liter) ins Gefäß 2 (5 Liter)

Zustand: [1,5,2,]

Umfüllen: Von Gefäß 2 (5 Liter) ins Gefäß 3 (3 Liter)

Zustand: [1,4,3,]

Umfüllen: Von Gefäß 3 (3 Liter) ins Gefäß 1 (8 Liter)

Zustand: [4,4,0,]

Ziel erreicht