

# Aufgabe 3: Telepaartie

Team-ID: 00587

Team-Name: Doge.NET

Bearbeiter dieser Aufgabe:  
Johannes von Stoephasius

20. November 2019

## Inhaltsverzeichnis

|          |                              |          |
|----------|------------------------------|----------|
| <b>1</b> | <b>Lösungsidee</b>           | <b>1</b> |
| 1.1      | Definitionen . . . . .       | 1        |
| 1.2      | Kernidee . . . . .           | 1        |
| 1.3      | 1. Teilalgorithmus . . . . . | 2        |
| 1.4      | 2. Teilalgorithmus . . . . . | 2        |
| 1.5      | Hauptalgorithmus . . . . .   | 2        |
| <b>2</b> | <b>Umsetzung</b>             | <b>2</b> |
| <b>3</b> | <b>Beispiele</b>             | <b>2</b> |
| <b>4</b> | <b>Quellcode</b>             | <b>2</b> |

## 1 Lösungsidee

### 1.1 Definitionen

Ein Zustand ist definiert als Menge von Behältern, wobei jedem Behälter eine positive oder neutrale natürliche Zahl zugeordnet werden kann, die der Anzahl an Bibern des Gefäßes entspricht. Weiter können die Behälter untereinander getauscht werden, da die Konstellation die selbe bleibt. Deshalb werden die Biber-Anzahlen in einer Liste gespeichert, die stets aufsteigend sortiert ist.

### 1.2 Kernidee

Die Grundidee der Lösung basiert auf der Idee, vom Endzustand den idealen Weg zu den Anfangszuständen zu finden. In diesem Fall heißt das, dass ein Baum aufgebaut wird, wobei die Knoten einzelne Zustände symbolisieren. Die Kinder eines Knotens sind alle möglichen Zustände, die mit dem 2. Nebenalgorithmus gefunden werden können. Der Baum hat nicht wie üblich nur einen Kopf, sondern mehrere, die vom 1. Nebenalgorithmus gefunden werden. Die Köpfe und damit der Return-Wert des Nebenalgorithmus sind alle Möglichen Zustände, bei denen genau ein Behälter 0 Elemente enthält. Nun werden mit dem 2. Teilalgorithmus alle Zustände gefunden, aus denen der aktuelle Zustand gebildet werden kann, woraus die neue Generation entsteht. Von den neuen Zuständen werden alle entfernt, die bereits gefunden wurden, sodass keine Dopplungen auftreten können. Auch auf diese neuen Zustände werden alle möglichen Operationen angewendet, wobei dies so lange wiederholt wird, bis keine neuen Zustände gefunden werden. Die Eltern dieser Generation ohne Zustände sind die Zustände, die am meisten Schritte brauchen, um in einen zulässigen Endzustand überführt zu werden.

### 1.3 1. Teilalgorithmus

### 1.4 2. Teilalgorithmus

Der Algorithmus findet für einen Zustand  $Z$  alle Zustände für die gilt, dass wenn auf sie alle möglichen Telepaartien angewendet eines der Ergebnisse  $Z$  ist. Dafür wird jede Biber-Anzahl mit jeder anderen Biber-Anzahl verglichen. Ist die erste Anzahl größer als 0 und durch 2 teilbar, dann ist es möglich, dass auf diese beiden Behälter eine Telepaartie angewendet wurde. Um diese umzukehren wird die Anzahl im ersten Behälter addiert und die Differenz zum 2. Behälter addiert.

### 1.5 Hauptalgorithmus

Zuerst wird mit dem 1. Teilalgorithmus eine Liste an Zuständen gebildet, die alle möglichen Endzustände abbilden. Diese Liste, im Folgenden VaterListe genannt, enthält am Anfang jeder der folgenden Iterationsschritte die Zustände, deren Kinder gebildet werden. Weiter existiert eine Liste an Zuständen, im Folgenden AlleZuständeListe genannt, die am Ende jedes Iterationsschritts jeden bereits gefundenen Zustand enthält, weshalb alle Zustände aus VaterListe in sie kopiert werden. Die nun beginnende Iteration erzeugt mit dem 2. Teilalgorithmus angewendet auf jeden Zustand in VaterListe eine Liste an Zuständen; die nächste Generation. Von diesen werden nun all die Zustände entfernt, die mehrfach existieren, um dopplungen zu vermeiden. Nun werden all die Zustände entfernt, die bereits in der AlleZuständeListe vorhanden sind. Die nun übrigen Elemente werden zur AlleZuständeListe hinzugefügt und bilden die neue VäterListe. War jedoch die Anzahl der übrigen Elemente gleich 0, so sind die Eltern dieser leeren Generation die Zustände, die die meisten Schritte brauchen, um einen Behälter zu leeren.

## 2 Umsetzung

## 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.