

## Junioraufgabe 1: Parallelen

### J1.1 Lösungsidee

Zunächst stellt sich die Frage, welche Antwort man erwarten sollte. Hat Martin recht? Manuelles überprüfen beim Gedicht zeigt: Er hat recht. Bei allen Wörtern der ersten Hälfte kommt man bei dem Wort „verschlang“ heraus. Erst bei dem zweiten Wort in der zweiten Hälfte kommt man zum ersten Mal zu einem anderen Wort, nämlich „Ewigkeit“.

Dies liegt jedoch nicht an der speziellen Struktur des Gedichts und dem Genie des Autors, sondern ist eine einfache Konsequenz daraus, dass bei hinreichender Textlänge es sehr wahrscheinlich ist, dass die „Pfade“, die von zwei verschiedenen Wörtern ausgehen, sich irgendwann vereinen. Mit „Pfad“ ist an dieser Stelle die Abfolge von Wörtern gemeint, zu denen man bei der Anwendung des Verfahrens springt. Die gleiche Überlegung kann auch für die Konstruktion eines Kartentricks genutzt werden<sup>1</sup>, hier sind auch mathematische Überlegungen zu Wahrscheinlichkeiten von Kollisionen gegeben.

Als Beispiel hier die ersten beiden Strophen des Gedichts. Alle Wörter, die man erreicht, wenn man mit dem ersten Wort „Es“ beginnt, sind rot markiert.

Es gingen zwei Parallelen  
ins Endlose hinaus,  
zwei kerzengerade Seelen  
und aus solidem Haus.

Sie wollten sich nicht schneiden  
bis an ihr seliges Grab:  
Das war nun einmal der beiden  
geheimer Stolz und Stab.

...

Nun noch einmal die ersten beiden Strophen. Diesmal wurde die Methode aber für die nächsten drei Wörter angewandt. Während es in der ersten Strophe noch Unterschiede bei den markierten Wörtern gibt, sind sie in der zweiten am Ende bereits alle identisch.

Es gingen zwei Parallelen  
ins Endlose hinaus,  
zwei kerzengerade Seelen  
und aus solidem Haus.

Sie wollten sich nicht schneiden  
bis an ihr seliges Grab:  
Das war nun einmal der beiden  
geheimer Stolz und Stab.

...

---

<sup>1</sup>Jeffrey C. Lagarias, Eric Rains, Robert J. Vanderbei: The Kruskal Count, 2001, <https://arxiv.org/abs/math/0110143>

## Einfache Lösung

Die einfache Lösung des Problems ist, das beschriebene Verfahren für jedes Wort in der ersten Hälfte des Gedichtes durchzuführen und dabei zu prüfen, ob man jeweils auf das gleiche Ergebnis kommt wie das Resultat beim ersten Wort.

Dabei muss insbesondere beachtet werden, dass nach der Aufgabenstellung nur die Buchstaben gezählt werden sollen. Die Satzzeichen müssen also vorher herausgefiltert werden oder anderweitig bei der Zählung der Zeichen ignoriert werden.

Die einfache Lösung hat im schlechtesten Fall quadratische Laufzeit abhängig von der Anzahl der Wörter im Gedicht.

## Lineare Lösung

Eine Verbesserung kann erzielt werden indem man realisiert, dass das Verfahren abgebrochen werden kann, sollte man auf ein Wort stoßen, dass Teil eines Pfads eines vorigen Wortes ist. In diesem Falle weiß man bereits, dass das momentan untersuchte Wort auf dem gleichen Wort enden wird wie die vorigen.

Diese Erkenntnis kann auf unterschiedliche Weise genutzt werden. Eine Möglichkeit ist beispielsweise, die Wörter während des Verfahrens zu markieren. Dazu bietet sich ein Array an, das für jedes Wort speichert, ob sein Pfad zum gleichen Wort führt wie der Pfad des ersten Wortes. Wenn das Verfahren dann für das erste Wort ausgeführt wird, werden alle besuchten Wörter markiert. Bei allen folgenden Wörtern wird auch so verfahren, nur dass gestoppt wird, wenn auf ein bereits markiertes Wort gestoßen wird. Sollte man bei einem der folgenden Wörter das Ende des Gedichts erreichen, ohne auf ein markiertes Wort gestoßen zu sein, hat man das erste Wort gefunden, für das die Behauptung nicht gilt.

Dieses Verfahren sorgt dafür, dass jedes Wort quasi nur einmal besucht wird und nicht für jedes Wort das gesamte Verfahren ausgeführt werden muss. Zu überprüfen, ob ein Wort bereits markiert ist, erfordert einen einfachen Array-Speicherzugriff und geht sehr schnell.

## J1.2 Alternative Lösungsideen

Das gleiche Verfahren kann auch von hinten nach vorne angewendet werden, man erhält somit dann etwas eher ein paar Wörter, deren Pfad auf verschiedenen Wörtern endet. Vor allem aber erstmal langweilige, wie dem letzten und vorletzten Wort. Daher sollte hier das Verfahren bei einem Gegenbeispiel nicht direkt abgebrochen werden.

Auch denkbar ist aus dem Gedicht einen Graphen explizit zu konstruieren und dann zu prüfen, ob der Graph zusammenhängend ist.

## J1.3 Beispiele

Bei dem gegebenen Gedicht funktioniert das Verfahren für alle Wörter bis zum Wort „wie“ in der vorletzten Strophe. Ausgehend von „wie“ landet man beim letzten Wort des Gedichts „Seraphim“, während alle vorigen Wörter zu „verschlang“ in dem vorletzten Vers der letzten Strophe führen.

**Algorithmus 1 Lösung**


---

```

for  $i$  in  $[0, \text{Anzahl Worte})$  do
    markiert( $i$ )  $\leftarrow$  false
end for
for  $i$  in  $[0, \text{Anzahl Worte})$  do
     $k \leftarrow i$ 
    while  $k < \text{Anzahl Worte}$  do
        if markiert( $k$ ) then
            break
        else
            markiert( $k$ )  $\leftarrow$  true
        end if
         $k \leftarrow k + \text{wortLaenge}(\text{gedicht}[k])$ 
        if  $i > 0$  und  $k \geq \text{Anzahl Worte}$  then
            gib aus "Gegenbeispiel gefunden"
        end if
    end while
end for

```

---

Ein gutes Beispiel zum Testen des Programmes könnte der folgende oder ein ähnlicher String sein: „aa bb aa bb aa bb aa bb aa bb“. Bei diesem ist das Gegenbeispiel bereits das zweite „Wort“ (aa  $\rightarrow$  aa  $\rightarrow$  aa  $\rightarrow$  aa  $\rightarrow$  aa, bb  $\rightarrow$  bb  $\rightarrow$  bb  $\rightarrow$  bb  $\rightarrow$  bb).

Auch möglich ist das Testen auf anderen Gedichten oder gar Romanen; eine gute Quelle hierfür stellt Projekt Gutenberg dar. Neben den obigen Überlegungen würden auch solche Tests zeigen, dass Christian Morgenstern nicht allein steht mit Texten, für die das Verfahren funktioniert.

## J1.4 Bewertungskriterien

Die Bewertungskriterien vom Bewertungsbogen werden hier erläutert (Punktabzug in []).

- [−1] **Lösungsverfahren fehlerhaft**  
Zur Bestimmung der Sprunglänge sollen nur Buchstaben gezählt werden, keine Satzzeichen. Der Gedankenstrich im Text ist kein eigenes Wort.
- [−1] **Implementierung fehlerhaft**  
Die Implementierung sollte die beschriebene Lösung korrekt umsetzen. Wenn im Programm z. B. immer  $n$  Wörter übersprungen werden (also immer  $n + 1$  Wörter weitergegangen wird), statt  $n$  Wörter weiter zu gehen, ist das nur in Ordnung, wenn das auch in der Dokumentation so beschrieben ist.
- [−1] **Ausgabe schlecht nachvollziehbar**  
Die Ausgabe des Programms sollte erkennbar machen, dass die von Wörtern der ersten Hälfte des Gedichtes ausgehenden Pfade alle bei dem Wort „verschlang“ enden. Außerdem soll nachvollziehbar sein, wie von einem einzelnen Wort das End-Wort erreicht wird.
- [−1] **Ergebnis unzureichend begründet**  
Das (positive) Ergebnis für das Gedicht „Die zwei Parallelen“ muss angegeben werden. Außerdem soll es durch die Angabe einer hinreichenden Zahl von „Pfaden“, die alle zu demselben Wort führen, begründet werden. Es ist nicht nötig, für alle Wörter aus der ersten Hälfte des Gedichts die Pfade anzugeben.