# Kekse, ich brauch Kekse!

Bearbeiter/-innen dieser Aufgabe:
Lisa Bergmann

16. Februar 2023

## Inhaltsverzeichnis



## 1 Lösungsidee

Das Problem, möglichst schnell an einer Warteschlange vorbeizudrängeln lässt sich mit einem Programm lösen, welches in einer Liste die am weitesten entfernte Lücke innerhalb der nächsten 11 Plätze findet. Das liegt daran, da es laut Aufgabenstellung möglch ist an 10 Personen gleichzeitig vorbeizulaufen. Somit ist, wenn man gedanklich die Positonen numeriert und auf Position p=0 startet, und eine Lücke auf Position 11 ist, möglich als nächstest in diese Lücke vorzudrängeln, da man bei diesem Vorgang an genau 10 Positionen vorbeiläuft. Da dies die maximal mögliche Entfernung eines Überholvorganges ist, muss bei jedem Schritt zu Beginn die Position p+11 nach einer Lücke übeprüft werden. Wenn sich an dieser Stelle keine Lücke befindet, geht das Programm von der Position p+11 in Richtung p durch die Warteschlange bis zur nächsten Lücke. Im Sonderfalle, dass es in der Position von p bis p+11 keine Lücke gibt, wird das Programm gestoppt und ein Fehler ausgegeben. Ansonsten wird die nächst günstigste Lücke als neues p definiert, ein Vordrängelvorgang ist abeschlossen und die nächste kann beginnen. Wenn p die letzte Lücke also der letzte Wert der Liste ist, ist man am Anfang der Warteschlange angekommen und hat sein Ziel erreicht. Die Anzahl der abgeschlossenen Überholvorgänge kann ausgegeben werden.

## 2 Umsetzung

Umgesetzt kann diese Programmidee in Python werden. Die Warteschlange ist dabei eine Zahlenfolge, wobei man an Stelle 0 startet und an der letzten Lücke, Stelle n ankommen will. Als Eingabe in das Programm erhält man zu Beginn die Stellen der Lücken, welche von klein nach groß geordnet in einer Liste 'schlange' gespeichert werden. Die Aktuelle Position p und der Counter der Überholvorgänge 'loesung' werden auf null gesetzt. Nun wird solange 'p' kleiner als der letzte Wert der Liste, also die Zielstelle, ist, eine Funktion 'vordraengeln' ausgeführt. Diese Funktion sucht beginnend bei 'p'+11 in Richtung 'p' rekursiv nach einer Lücke. Ist diese gefunden bricht die Rekursion ab und die Stelle der Lücke wird als neues 'p' bestimmt. Der Counter 'loesung' wird um eins erhöht. Wird keine Lücke gefunden wird eine Fehlermeldung ausgegeben. Wenn 'p' schließlich gleich dem letzten Wert der Liste ist, ist das Vordrängeln abgeschlossen und der Wert des Counters ann ausgegeben werden.

## 3 Beispiele

Ausgabe der Beispielschlange 'schlange.txt': 'Es werden 327 Vordrängelaktionen benötigt.'

Kekse, ich brauch Kekse!

## 4 Quellcode

```
schlange=eingabe.split() #textdatei wird gelesen und in Liste eingeschrieben
p = 0 #Anfangsposition 0 gesetzt
loesung = 0 #Variable fuer Anzahl der Ueberholvorgaenge 0 gesetzt

def vordraengeln(aktuelleposition, testposition, warteschlange):
    #Funktion, welche die bestmoegliche neue Position nach einem Ueberholvorgang ermittelt
    if str(testposition) in warteschlange:
    #wenn aktuell ueberpruefte Position in Liste,
    #also Luecke, return neue position
        return testposition
    else: #ansonsten naechst kleinere Position ueberpruefen, rekursion bis Luecke gefunden
        testposition -=1
        if testposition <= aktuelleposition:
            #Wenn Keine neue Luecke innerhalb der naechsten 11 gefunden, abbruch
            print('Es ist nicht moeglich zu wiederholen')
            exit()
        return vordraengeln(p, testposition, warteschlange)

while p < int(schlange[-1]): #solange p nicht an letzter Stelle steht
    versuche = p+11
    p = vordraengeln(p, versuche, schlange) # definiere p als die neue Luecke
    loesung += 1 #der Counter der Loesung wird um eins erhoeht

print('Es werden ' + str(loesung) + ' Vordraengelaktionen benoetigt.')
#Ausgabe des Erebnisses
```