Aufgabe 2: Nummernmerker

Team-ID: 00619

Team-Name: ncrypted

Bearbeiter/-innen dieser Aufgabe:
Oliver Schirmer

22. November 2019

Inhaltsverzeichnis

Lösungsidee

Zur Lösung dieses Problems wird ein Gready-Algorithmus gewählt. Dieser beginnt jedoch beim Ende der Zahl Blöcke zu bilden. D.h. es gibt einen Index, der das Ende eines Blockes festlegt. Dabei wird immer mit maximaler Blockgröße begonnen und geprüft ob die aktuelle Ziffer, mit der der Block beginnen würde eine 0 ist. Ist dies der Fall, wird die Blockgröße bis zur Größe zwei vermindert. Beginnt der Block in diesem Fall immer noch mit einer 0, wird der Block auf die in der Situation maximal mögliche Größe gesetzt, da der aktuelle Block in jedem Fall mit 0 beginnen muss, jedoch der nächste Block damit möglichst nah an eine zu 0 verschiedene Zahl gebracht wird und somit die Anzahl an mit 0 beginnenden Blöcken minimiert wird. Wird beim Überprüfen der aktuellen Blockgröße zu Beginn eine zu 0 verschiedene Zahl gefunden, wird der Block gespeichert, der Index auf den Blockanfang gesetzt und es wird der nächste gebildet.

Umsetzung

Die Zahl die in Blöcke zu zerlegen ist wird nicht als Zahl, sondern als String gespeichert, da die einzelnen Ziffern einfacher zu überprüfen sind und zudem weniger Speicherplatz benötigt wird. Die einzelnen Blöcke werden lediglich als Indexes in einer Liste gespeichert, die angeben, an welcher Stelle der jeweilige Block beginnt. Beim späteren Hinzufügen der Trenner sind die Indexes danach ungültig, da sich der jeweilige Index nach jedem Einfügen um einen nach rechts verschiebt, was in diesem Kontext jedoch irrelevant ist, da keine spätere Nutzung nötig ist.

Der tatsächliche Algorithmus hat keine bestimmte Anzahl an Durchläufen, weshalb solange Blöcke gebildet werden müssen, bis der Index kleiner als fünf ist, da in diesem Fall bei der maximalen Blockgröße von vier noch gewährleistet ist, dass der letzte gebildete Block noch eine Mindestgröße von zwei aufweist.

Beispiele

Eingabe	Ausgabe
005480000005179734	005-4800-0000-517-9734
03495929533790154412660	034-9592-9533-7901-5441-2660
5319974879022725607620179	531-9974-8790-227-2560-7620-179
9088761051699482789038331267	9088-7610-5169-9482-7890-3833-1267
01100000001100010011111111011011	011-0000-0001-1000-100-1111-1110-1011
10000	10-000

Die ersten fünf Beispiele sind der BwInf-Webseite entnommen. Das letzte Beispiel verdeutlicht nochmals, dass selbst beim Wiedererhöhen auf die maximale Blockzahl von vier, aufgrund von Nullen an jeder möglichen Anfangsposition, die Blockgröße des ersten Blocks nicht unter zwei sinkt. Es wird also nicht auf die maximale Blockgröße, sondern die in der aktuellen Situation maximal möglichen Blockgröße gesetzt.

Quellcode

```
package de.ncrypted.merker;

public class Merker {

    private static String number;
    private static String numberSplit;

    public static void main(String[] args) {
        // read number ...

        executeAlgorithm();

        System.out.println("---Ergebnis---");
        System.out.println(numberSplit);
    }

    private static void executeAlgorithm() {
        int idx = number.length();
        List<Integer> splitPoints = new ArrayList<>();
        while (idx > 4) {
            for (int i = 4; i >= 2; i--) {
                int splitIdx = idx - i;
                if (number.charAt(splitIdx) == '0') {
                    if (i != 2) {
                        continue;
                    }
                    splitIdx -= 2;
                }
                if(splitIdx < 2) {
                    splitIdx = 2;
                }
                splitPoints.add(splitIdx);
                idx = splitIdx;
                break;
            }
        }
        StringBuilder builder = new StringBuilder(number);
```

```
            for (int i = 0; i < splitPoints.size(); i++) {
                builder.insert(splitPoints.get(i), "-");
            }
            numberSplit = builder.toString();
        }
    }
```