Tag 3: Binäre Diagnostik

Teil 1

Für Teil 1 reicht es aus, wenn man die Gamma-Rate bestimmt, da die Epsilon-Rate das bitweise Inverse der Gamma-Rate ist.

Um die Gamma-Rate zu bestimmen, muss man durch alle Zeilen der Eingabe und alle Stellen der Zahlen gehen und die Einsen zählen. Ein Methode, die hier wertvoll ist, ist `toCharArray()` der String-Klasse:

```
char[] chars = line[0].toCharArray();
```

So erhält man ein Array aus `char`-Werten, so kann man stellenweise über die Werte der Zeilen iterieren und die Einsen zählen. Bechten sollte man, dass die Werte in der Beispieldatei kürzer sind, als die im Input, hier muss flexibel programmiert werden, z.B. indem man mal schaut, wie lange die erste Zahl ist:

```
int numBits = list.get(0)[0].toCharArray().length;
```

Teil 2

In Teil zwei kann man die ArrayList verkleinern, indem man sich bei jedem Durchlauf merkt, welche Zeilen gelöscht werden sollen:

```
// Erstelle eine Kopie der InputListe mit dem Namen myList
ArrayList<String[]> myList = new ArrayList<String[]>(list);
// Erstelle eine Liste, die die zu löschenden Elemente aufnimmt
ArrayList<String[]> toDelete = new ArrayList<String[]>();


// Für jedes Bit...

// Finde raus, welche zeilen gelöscht werden müssen
// füge diese in die toDelete Liste ein:
[...]
          toDelete.add(line);
[...]
          // to Delete aus myList löschen
          myList.removeAll(toDelete);
          // toDelete leeren für den nächsten Bit-Durchlauf
          toDelete.clear();
          // Bit-Durchläufe abbrechen, wenn es nur noch
          // einen Wert gibt
          if (myList.size() == 1) {
                break;
```

```
            }
[...]
```

From:
https://wiki.qg-moessingen.de/ - **QG Wiki**

Permanent link:
https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:java:aoc:aoc2021:day3:start?rev=1638806227

Last update: **06.12.2021 16:57**