

001 Problem Solving: Rummy Numbers

Maximilian Noppel¹

¹max [at] vspace.one

Zusammenfassung

Wir lösen wieder ein Teilproblem von Rummy. Im Vergleich zum 005 CodeGolfing geht es dieses mal nur darum eine Lösung implementieren. Die Länge des Sourcecodes ist dabei egal. Außerdem schauen wir uns dieses mal nicht die Farben der Karten sondern die Zahlen an.

Schwierigkeit: 

EINFÜHRUNG

Rummy ist ein relativ einfaches Kartenspiel. Jede Karte hat ein Zahl von 1 bis 13 und eine von vier Farben (schwarz, blau, gelb und rot)¹. Das Ziel vom Rummy ist es Karten zu Reihen aus mindestens drei Karten zusammenzulegen. Dabei müssen entweder alle Karten die gleich Farbe haben und aufsteigenden Zahlen oder alle Karten die gleiche Zahl aber unterschiedliche Farben. Keine Karte darf dabei übrigbleiben oder in einer Zweier-Reihe sein. Beim letzten CodeGolfing (005) haben wir dieses Problem für Karten gelöst, die alle die gleiche Zahl hatten. Wir mussten also nur Dreier/Vierereihen aus verschieden-farbigen Karten bilden.

Nun lösen wir das Problem für Zahlen, die alle die gleiche Farbe haben. Hier können wir deutlich längere Reihen bilden. Zum Beispiel [1] [2] [3] [4] [5] [6] [7] . . . Außerdem können wir nach der höchsten Zahl (13) wieder von vorne anfangen. Die Reihe [12] [13] [1] ist also eine gültige Reihe.

AUFGABE

Die Aufgabe ist also wie folgt: Gegeben seien 13 Zahlen zwischen 0 und 100. Jede dieser Zahlen repräsentiert die entsprechenden Anzahl an Karten. Die erste Zahl gibt also die Anzahl an einsern, die zweite die Anzahl an zweiern und so weiter an. Diese 13 Zahlen werden über die Kommandozeile an das Programm übergeben. Für Python beispielsweise so:

```
$ python your_script.py 4 9 34 75 10 90 45 23 7 76 25 19 95
```

Oder für C Programme so

```
$ ./main 4 9 34 75 10 90 45 23 7 76 25 19 95
```

Das Programm soll ausgeben ob die gegeben Karten ordnungsgemäß abgelegt werden können. Also ob es eine Ablage gibt, so dass keine Karte alleine oder in einer Zweierreihe liegen bleibt. Wie die Ablage tatsächlich aussieht ist nicht interessant. Natürlich muss jede Karte abgelegt werden.

Das Programm signalisiert das Ergebnis *Ablage möglich* über den Rückgabewert 0 und *Ablage nicht möglich* über einen Wert $\neq 0$, z.B. -1

UPLOAD

Da ich keine Lust haben meinen Code am Ende wieder so zu verunstalten und da beim letzten CodeGolfing die Beteiligung an der Abschlussveranstaltung so gering war, dürft ihr die Aufgabe nach Belieben lösen und eure Lösung ins Wiki packen. Link: <https://wiki.vspace.one/doku.php?id=treffen:problemsolving001>

¹Eigentlich gibt es noch zwei Joker, aber die ignorieren wir hier mal.