

10. Schmutzige Tricks im Einheitskostenmaß, 10 Punkte

Gegeben:

$$x = (x_1, \dots, x_n)$$

$$y = (y_1, \dots, y_n)$$

$$z = (z_1, \dots, z_n)$$

$$M = \max\{|x_i|, |y_i|, |z_i| \mid i = \{1, \dots, n\}\}$$

$$u > 4M$$

$$a = \sum_{i=1}^n u^{n-1} x_i$$

$$b = \sum_{i=1}^n u^{n-1} y_i$$

$$c = \sum_{i=1}^n u^{n-1} z_i$$

zu Zeigen:

$$(a + b = c) \Leftrightarrow x + y = z$$

a.) \Leftarrow - Einfach einsetzen...

$$\begin{aligned} a + b &\stackrel{Def}{=} \sum_{i=0}^n x_i u^{n-i} + \sum_{i=0}^n y_i u^{n-i} \\ &= \sum_{i=0}^n (x_i + y_i) u^{n-i} \\ &\stackrel{Def}{=} \sum_{i=0}^n z_i u^{n-i} \\ &= c \end{aligned}$$

c.) \Rightarrow

Wissen: $a + b = c$

Idee: $\Rightarrow a + b = c \pmod{u}$, also irgendwie versuchen die u -Werte zu lösen.

$$\begin{aligned} a + b &= u \left(\sum \dots \right) + (x_n + y_n) \\ c &= u \left(\sum \dots \right) + (z_n) \\ \Rightarrow x_n + y_n &= z_n \pmod{U} \end{aligned}$$

11. Programmieren einer Registermaschine, 10 Punkte

Pseudocode

```
1 in (n, x[1,...,n])
2   max = x[1]
3   tmp = x[1]
4   for i = 2, i < n, i++
5     tmp = x[i]
6     if tmp > max
7       tmp = max
8   return max
```

RAM

```
1 ; r0 = n
2 ; r1-r5 sind frei (siehe letztes Tutorium)
3 ; r6-r(n+5) enthalten die Eingabe
4 r1 = 1 ; max_i
5 r2 = r6 ; max
6 r3 = r6 ; tmp
7 r4 = r6 ; i
8 r5 = 0 ;
9 loop:
10   r4 = r4 + 1 ; ++i
11   r5 = r0 + 6 ; abbruch wenn n+6-i
12   r5 = r5 - r4
13   GZ r5, halt
14
15   r3 = r3 + (r4) ; tmp aktualisieren
16   r5 = r3 - r2
17   GGZ R5, newmax
18   GOT0 loop
19 newmax:
20   r2 = r3 ; neues max setzen
21   r1 = r4 - 5 ; neues max_i setzen
22   GOT0 loop
23 halt:
24   HALT
```

Laufzeitanalyse

EKM: In der Summe $\Theta(n)$

LKM: $\mathcal{O}(nL)$

L = längster vorkommender Eintrag in einem Register.

Speicherbedarf

EKM: $\Theta(n)$ - wir benutzen $\Theta(n)$ Register

LKM: $\mathcal{O}(nL)$ laut Vorlesung

$\Rightarrow n$ Register mit Inhalt $\Omega(L)$