Open Class – Sieben Wunder der Informatik

Prof. Dr. Juraj Hromkovič

Departement Informatik

Lösungsvorschläge für die Übungsaufgaben – Blatt 4

Zürich, 14. August 2008

Lösung zu Aufgabe 11

Unter Anwendung des Distributivgesetzes schreiben wir das Polynom

$$a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

in die Form

$$(((a_4 \cdot x + a_3) \cdot x + a_2) \cdot x + a_1) \cdot x + a_0$$

um. Dieser Formel entnehmen wir sofort, dass vier Multiplikationen und vier Additionen ausreichen.

Lösung zu Bonus-Aufgabe 4

Wir führen das gegebene Problem zurück auf das Problem, ein Polynom $(n-1)$-ten Grades mit höchstens $n-1$ Multiplikationen und $n-1$ Additionen auszuwerten.
Wenn es uns gelingt, den Wert

$$t := a_nx^{n-1} + a_{n-1}x^{n-2} + \ldots + a_3x^2 + a_2x + a_1$$

mit nur $n-1$ Multiplikationen und $n-1$ Additionen zu bestimmen, so können wir mit Hilfe einer weiteren Multiplikation und einer weiteren Addition auch

$$t \cdot x + a_0 = a_nx^n + a_{n-1}x^{n-1} + \ldots + a_2x^2 + a_1x + a_0$$

bestimmen.
Den hieraus resultierenden Algorithmus kann man etwa mit folgendem Pseudo-Code angeben, der (nach x) die Koeffizienten in der Reihenfolge a_n, …, a_0 einliest:

Lies x ein.
$y \leftarrow 0$
Solange noch ein Koeffizient a eingelesen werden kann,
  $y \leftarrow y \cdot x$
  $y \leftarrow y + a$
Gib y aus.

Lösung zu Aufgabe 12

Bei einer Rechengeschwindigkeit von 10^9 Operationen pro Sekunde können während eines Zeitraums von 10^{18} Sekunden maximal 10^{27} Operationen ausgeführt werden.
Wir können also Eingabegrössen bearbeiten wie folgt:

a)
$$\begin{aligned} 10 \cdot n^2 &\leq 10^{27} && \mid \div 10 \\ n^2 &\leq 10^{26} && \mid \sqrt{} \\ n &\leq 10^{13} \end{aligned}$$

b)
$$\begin{aligned} 50 \cdot n^3 &\leq 10^{27} && \mid \div 50 \\ n^3 &\leq 20 \cdot 10^{24} && \mid \sqrt[3]{} \\ n &\leq 10^8 \cdot \underbrace{\sqrt[3]{20}}_{>2.71441761} \\ n &\leq 271'441'761 \end{aligned}$$

c)
$$\begin{aligned} 2^n &\leq 10^{27} = 2^{27 \cdot \log_2 10} && \mid \log_2 \\ n &\leq 27 \cdot \underbrace{\log_2 10}_{\approx 3.32} \\ n &\leq 89 \end{aligned}$$

d)
$$\begin{aligned} n! &\leq 10^{27} \in (26!, 27!) \\ n &\leq 26 \end{aligned}$$

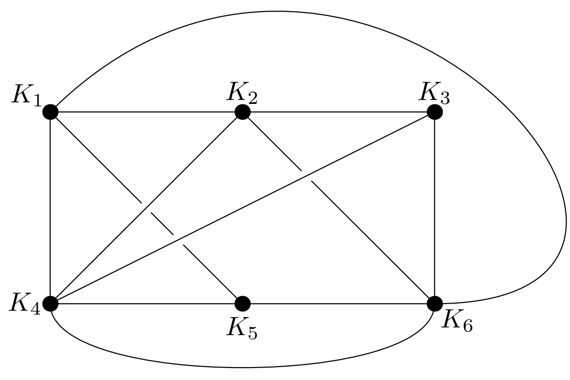
Es gilt nämlich:

$$\begin{aligned} 26! &= \quad 403'291'461'126'605'635'584'000'000 \qquad \text{und} \\ 27! &= 10'888'869'450'418'352'160'768'000'000 \quad . \end{aligned}$$

Der Wert 27! hat also bereits 29 Stellen, wohingegen 10^{27} die kleinste Zahl mit nur 28 Stellen ist. Folglich kann der Algorithmus A höchstens auf Eingaben der Grösse 26 praktisch arbeiten.

Lösung zu Aufgabe 13

Zunächst nummerieren wir die Kreuzungen:



Wir kodieren das Vorhandensein von Strassen in Ungleichungen wie folgt:

$$
\begin{array}{lll}
K_1 + K_2 \geq 1 & K_2 + K_3 \geq 1 & K_3 + K_6 \geq 1 \\
K_1 + K_4 \geq 1 & K_2 + K_4 \geq 1 & K_4 + K_5 \geq 1 \\
K_1 + K_5 \geq 1 & K_2 + K_6 \geq 1 & K_4 + K_6 \geq 1 \\
K_1 + K_6 \geq 1 & K_3 + K_4 \geq 1 & K_5 + K_6 \geq 1
\end{array}
$$

Die voranstehenden Ungleichungen drücken aus, dass die jeweilige Strasse beobachtet werden muss. Wir müssen nun noch ausdrücken, dass maximal drei Beobachter zugelassen sind:

$$K_1 + K_2 + K_3 + K_4 + K_5 + K_6 \leq 3$$

Gemeinsam drücken die vorangegangenen 13 Ungleichungen also genau die gegebene Instanz des Aufsichtsproblems aus.
Man beachte, dass das Strassennetz *nicht* mit drei Aufsehern überblickt werden kann. (Hiervon überzeugt man sich, indem man die zwanzig Möglichkeiten, die es gibt, die Aufseher zu positionieren, einzeln durchspielt.) Infolgedessen erlaubt das beschriebene System von 13 Ungleichungen auch keine 0-1-Lösung.