

Angewandte Mathematik

Kontakt: simon.fischer@uni-flensburg.de, 0461 8053109

Mittwochs 18 -19 Uhr
an Schultagen



Rätsel

Aufgabe

Ordne die Zahlen von 1 bis 15 so, dass die Summe von je zwei benachbarten Zahlen eine Quadratzahl ergibt.



Rätsel

Aufgabe

Ordne die Zahlen von 1 bis n so, dass die Summe von je zwei benachbarten Zahlen eine Quadratzahl ergibt.

Geht das für $n = 14$ und $n = 16$?



Graphentheorie Notation

Definition (Graph)

Ein Graph G ist ein geordnetes Paar (V, E) , wobei V eine Menge von "Ecken" (engl. vertex/vertices, dt auch Knoten) und E eine Menge von "Kanten" (engl. edges), bezeichnet. Graphen werden manchmal nach der Art der Kanten unterschieden. Z.B.

- ▶ E besteht aus ungeordneten Paaren $(v_1, v_2) \rightarrow$ *ungerichteter Graph*
- ▶ $E \subset V \times V \rightarrow$ *gerichteter Graph*
- ▶ Mehrfachkanten möglich, z.B. $E : V \times V \rightarrow \mathbb{N} \rightarrow$ *gerichteter Graph mit Mehrfachkanten*

Ecken heißen *benachbart*, wenn sie durch eine Kante verbunden sind.



Graphentheorie Notation

Definition (Weg)

Eine Folge von benachbarten Ecken heißt *Weg*.

- ▶ Ein Weg, der jede Kante genau einmal enthält heißt *Eulerweg* (auch *Eulertour*).
- ▶ Ein Weg, der jede Ecke genau einmal enthält heißt *Hamilton-Weg*.
- ▶ Ein Weg, der an seinem Startpunkt endet, heißt *Kreis*.
Entsprechend sprechen wir von *Eulerkreis* und *Hamilton-Kreis*.



Tupffehler

Beim Spiel *Tupffehler* müssen Wörter ineinander Umgewandelt werden, indem

- ▶ Ein Buchstabe durch einen anderen Ersetzt wird,
- ▶ ein Buchstabe wird hinzugefügt, oder
- ▶ ein Buchstabe wird gestrichen.

Durch mehrfache Anwendung dieser Transformationen entstehen *Wortketten*.

Die Mathe^{SH} Python AG versucht möglichst lange solche Wortketten in der deutschen Sprache zu finden.

Aufgabe

Modelliere das Problem als Graph.



Längste Wortketten

Zunächst werden nur die 33 Wörter:

'RUMPF', 'DAMPFES', 'KAEMPFE', 'STAMPFT', 'STRUMPF',
'DAMPF', 'DAEMPFE', 'SUMPf', 'DAMPFER', 'TRUMPFE',
'DAMPFST', 'DAMPFTE', 'STAMPFE', 'TRUMPFT',
'KAEMPFT', 'RUMPFES', 'DAMPFEN', 'DAMPFE', 'KRAMPF',
'KAMPF', 'TRIUMPF', 'DAMPFT', 'DUMPF', 'TRUMPFS',
'DUMPFEN', 'SUMPfES', 'STUMPFE', 'TRUMPF',
'DAEMPFT', 'TRIUMPH', 'DUMPFER', 'DUMPFES', 'STUMPF'
verwendet.

Aufgabe

Zeichne den Graphen für das Problem und finde eine Längste Wortkette

Aufgabe

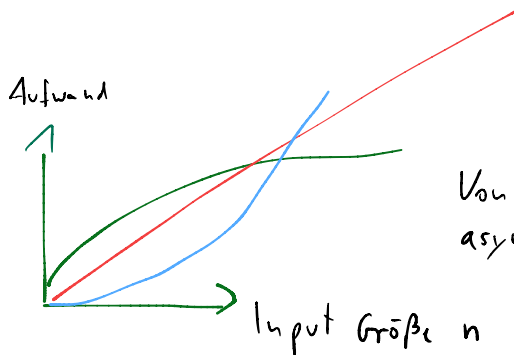
 *Beweise, dass es keine längere Wortkette geben kann.*



Algorithmische Komplexität

Definition (vage)

Die *algorithmische Komplexität* beschreibt, wie stark der zur Lösung eines Problems nötige Aufwand (Zeit, oder Speicher) mit der Größe des Inputs ansteigt.



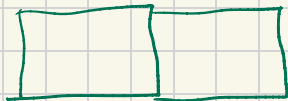
Von Interesse ist der
asymptotische Aufwand
 $n \rightarrow \infty$



Komplexität von Alltagsaufgaben

→ polynomiell

Rasenmäher



Zeit proportional zur Fläche (ohne Laser-Rasenmäher)

n Lineare Komplexität

Händeschütteln bei n Personen

$$\frac{n(n-1)}{2} \approx n^2$$

Quadratische Komplexität

N Karten sortieren

$$< n^2$$

Namen aus Telefonbuch mit n Einträgen

$$\log y/n)$$

logarithmische K.

→ exponentiell

Zahlschloss /
Passwort (länge n) raten

3 Zahnräder → 1000

4 Zahnräder → 10.000

2^n exponentielle
Kompl.

Komplexität von Computeraufgaben

Multiplikation von n stelligen Zahlen $\rightarrow n^2$

Gibt es einen Hamilton Weg im Graphen mit n Ecken
 $\rightarrow$ exponentiell? ($n!$)

Sind die Punkte s und t im Graphen verbunden? $\rightarrow n$

Landau Symbole

Mit den *Landau Symbolen* wird das asymptotische Wachstum von Funktionen beschreiben (also wie sehr sie für große inputs Wachsen)

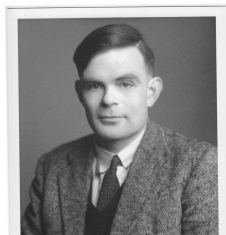
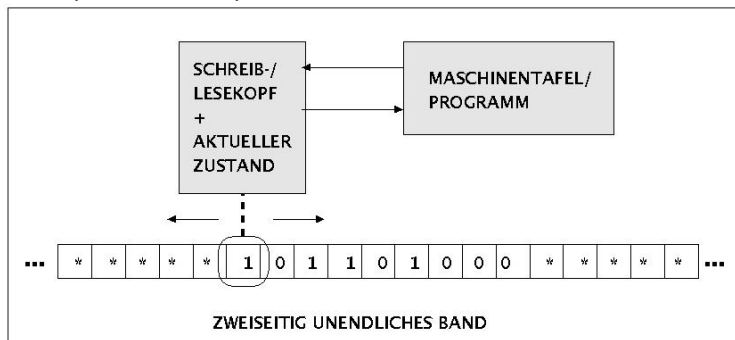
Bsp:

- ▶ $f \in o(g) \leftrightarrow \lim_{x \rightarrow \infty} \left| \frac{f(x)}{g(x)} \right| = 0$ $g=x, f=\text{sqrt}(x) = \sqrt{x}$
- ▶ $f \in O(g) \leftrightarrow \lim_{x \rightarrow \infty} \left| \frac{f(x)}{g(x)} \right| < \infty$ $g=x, f=2x$



Turingmaschinen

Eine (deterministische) *Turingmaschine* ist ein Theoretisches konzept eines Computer.



nicht deterministische Turingmaschinen

P vs NP

deterministisch polynomiell $\nearrow$ $\nwarrow$ nicht deterministisch

Eine *nicht deterministische Turingmaschine* A gibt bei einem Entscheidungsproblem (also eine Frage auf die die Antwort ja, oder nein ist) folgendes aus:

1. Ist die richtige Antwort Nein, so gibt A Nein aus
2. Ist die richtige Antwort Ja, so kann A Ja ausgeben.

„Raten und verifizieren“

