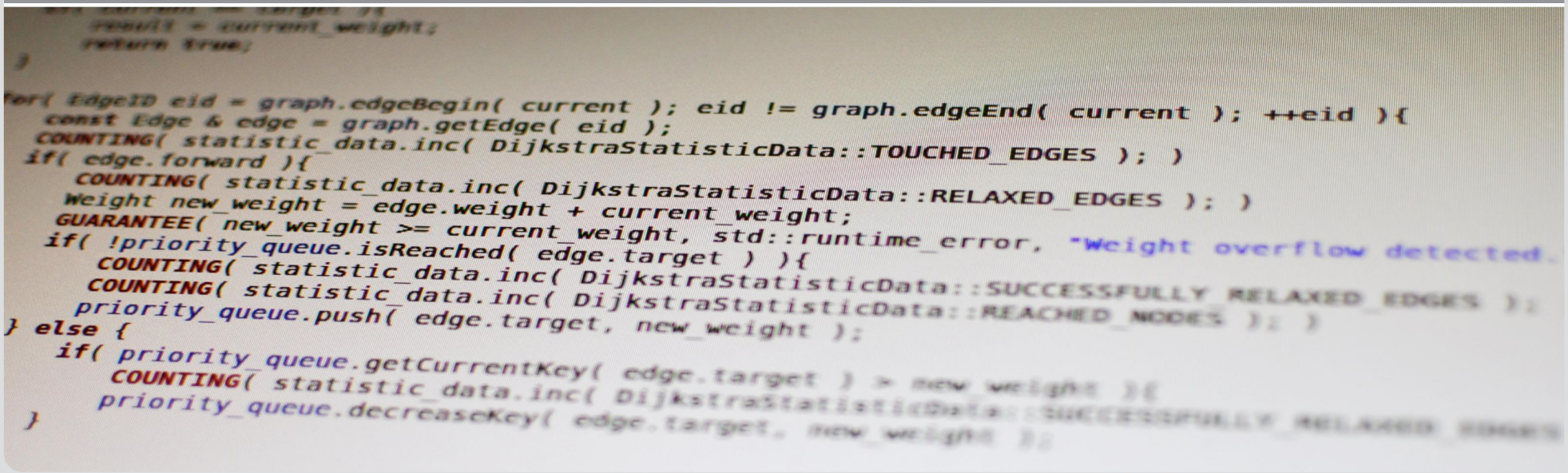
Karlsruher Institut für Technologie

Übung 7 – Algorithmen II

Moritz Laupichler, Nikolai Maas – *{moritz.laupichler, nikolai.maas}@kit.edu*

`https://algo2.iti.kit.edu/AlgorithmenII_WS23.php`

Institut für Theoretische Informatik - Algorithm Engineering



Themenübersicht

Karlsruher Institut für Technologie

Übungsinhalt

- Parametrisierte Algorithmen
- Besprechung Übungsblatt 3

Fixed-Parameter-Algorithmen

***fixed parameter tractable* (FPT)**

Warum Probleme parametrisieren?

- es gibt “schwierige” Probleme z.B. Minimum Independent Set
  → allgemeine Instanzen haben zu lange Berechnungszeit

- Kann man Spezialfälle eventuell effizient berechnen?
  → Identifizierung zusätzlicher Parameter k der Problemstellung
  → falls “Komplexität” in k steckt, effiziente Lösung für konstantes k möglich!

Fixed-Parameter-Algorithmen

Definition

- Ein Problem heißt fixed parameter tractable, wenn es eine Laufzeit

$$T(n,k) = \mathcal{O}(f(k) \cdot \mathrm{p}(n))$$

hat, mit $f(\cdot)$ berechenbar, $p(\cdot)$ Polynom.

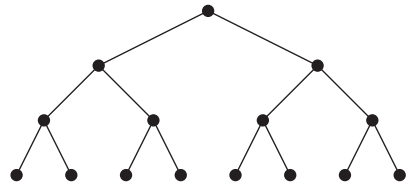
$f(\cdot)$ darf nicht von n abhängen und $p(\cdot)$ nicht von k

Fixed-Parameter-Algorithmen

Techniken

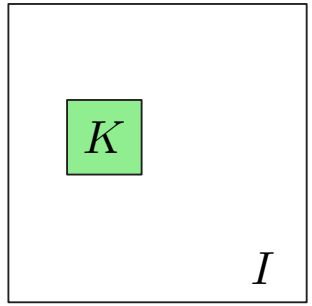
Tiefenbeschränkte Suche

- Aufzählen und Testen aller Möglichkeiten
  $\rightarrow$ mit geeignetem Suchbaum beschränkter Tiefe
  Tiefe ist abhängig von k



Kernbildung

- Probleminstanz auf (schwierigen) Problemkern reduzieren
- Problemkern mit anderer Technik lösen



Fixed-Parameter-Algorithmen

Schiebepuzzle

Problemstellung

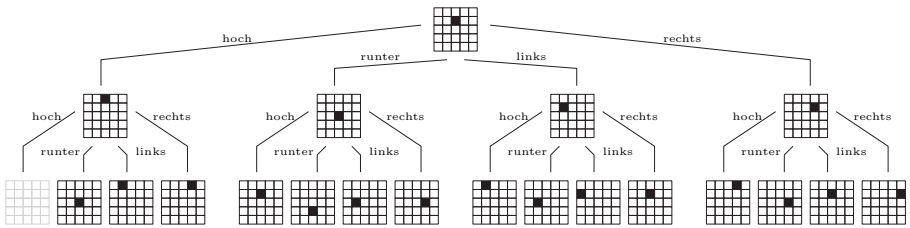
- gegeben $n \times n$ Schiebepuzzle, $k \in \mathbb{N}$
- entscheide, ob das Puzzle in $\leq k$ Zügen gelöst werden kann
  - das Puzzle ist gelöst, wenn die Teile sortiert sind
  - Loch wird pro Zug eine Position horizontal oder vertikal verschoben

24	8	13	12	20
11	2		17	21
7	15	14	19	5
6	10	3	9	1
4	23	11	18	22

Suchbaum-Algorithmus

- es gibt ≤ 4 Möglichkeiten in jedem Zug, k Züge
  - baue Suchbaum (Höhe k, Verzweigungsgrad ≤ 4)
    $\rightarrow$ Baumgröße $\mathcal{O}(4^k)$
  - teste jeden Knoten auf korrekte Lösung
    $\rightarrow$ Aufwand $\mathcal{O}(n^2) \in \mathcal{O}(poly(n))$

$\Rightarrow$ Gesamtaufwand: $\mathcal{O}(4^k n^2) \Rightarrow$ FPT



Ende!

Feierabend!