Fortgeschrittene Algorithmen

Laufzeitanalyse

- Landau-Symbol(e) – auch O-Notation
- Bester, durchschnittlicher, schlechtester Fall
- Armortisierte Analyse

Laufzeitanalyse

- Landau-Symbol(e) – auch O-Notation
- Bester, durchschnittlicher, schlechtester Fall
- Armortisierte Analyse

Kosten- / Laufzeitschätzung

Definition Kostenfunktion:

Gegeben sei ein Algorithmus A. Die Kostenfunktion $k : \mathbb{N} \rightarrow \mathbb{R}^+$ ordnet jeder Problemgröße n den Ressourcenbedarf (z.B. Anzahl Operationen) k(n) zu, die A zur Verarbeitung der Eingabe der Größe n benötigt.

Kosten- / Laufzeitschätzung

O-Notation: Drückt eine Größe aus, die eine bestimmte „Ordnung“ nicht übersteigt.

Definition O-Notation:

Es seien f und g zwei Kostenfunktionen. Wenn es eine Konstante c ∈ $\mathbb{R}$ und ein $n_0 \in \mathbb{N}$ gibt, so dass

f(n) ≤ c ·g(n) für alle n ≥ n_0 ,

dann schreiben wir $f \in O(g)$
(oder $f(n) \in O(g(n))$)

Laufzeitanalyse (Velocity)

O-Notation, wichtigste Regeln

- c = O(1)
- c * f(n) = O(f(n))
- O(f(n)) +O(f(n)) = O(f(n))
- O(O(f(n))) = O(f(n))
- O(f(n)) * O(g(n)) = O(f(n) * g(n))
- O(f(n)) + O(g(n)) = O(max{ f(n), g(n) })

Landau-Symbol(e) – auch O-Notation

Notation	Bedeutung	Anschauliche Erklärung	Beispiele für Laufzeiten
$f \in \mathcal{O}(1)$	f ist beschränkt.	f überschreitet einen konstanten Wert nicht (ist unabhängig vom Wert des Arguments n).	Feststellen, ob eine Binärzahl gerade ist Nachschlagen des n-ten Elementes in einem Feld in einer Registermaschine
$f \in \mathcal{O}(\log \log n)$	f wächst doppel-logarithmisch.	Bei Basis 2 erhöht sich f um 1, wenn n quadriert wird.	Interpolationssuche im sortierten Feld mit n gleichförmig verteilten Einträgen
$f \in \mathcal{O}(\log n)$	f wächst logarithmisch.	f wächst ungefähr um einen konstanten Betrag, wenn sich n verdoppelt. Die Basis des Logarithmus ist dabei egal.	Binäre Suche im sortierten Feld mit n Einträgen
$f \in \mathcal{O}(\sqrt{n})$	f wächst wie die Wurzelfunktion.	f wächst ungefähr auf das Doppelte, wenn sich n vervierfacht.	Anzahl der Divisionen des naiven Primzahltests (Teilen durch jede ganze Zahl $\leq \sqrt{n}$)
$f \in \mathcal{O}(n)$	f wächst linear.	f wächst ungefähr auf das Doppelte, wenn sich n verdoppelt.	Suche im unsortierten Feld mit n Einträgen (Bsp. Lineare Suche)

https://de.wikipedia.org/wiki/Landau-Symbole

Landau-Symbol(e) – auch O-Notation

Notation	Bedeutung	Anschauliche Erklärung	Beispiele für Laufzeiten
$f \in \mathcal{O}(n \log n)$	f hat super-lineares Wachstum.		Vergleichbasierte Algorithmen zum Sortieren von n Zahlen Mergesort, Heapsort
$f \in \mathcal{O}(n^2)$	f wächst quadratisch.	f wächst ungefähr auf das Vierfache, wenn sich n verdoppelt.	Einfache Algorithmen zum Sortieren von n Zahlen Selectionsort
$f \in \mathcal{O}(n^m)$	f wächst polynomiell.	f wächst ungefähr auf das 2^m -Fache, wenn sich n verdoppelt.	„Einfache“ Algorithmen
$f \in \mathcal{O}(2^n)$	f wächst exponentiell.	f wächst ungefähr auf das Doppelte, wenn sich n um 1 erhöht.	Erfüllbarkeitsproblem der Aussagenlogik (SAT) mittels erschöpfender Suche
$f \in \mathcal{O}(n!)$	f wächst faktoriell.	f wächst ungefähr auf das $(n+1)$-Fache, wenn sich n um 1 erhöht.	Problem des Handlungsreisenden (mit erschöpfender Suche)
$f \in A(n)$	f wächst wie die modifizierte Ackermannfunktion.		Problem ist berechenbar, aber nicht notwendig primitiv-rekursiv

Kostenfunktion → Präzisierung

- Erlaubte Arbeitsschritte
- (abstrakte) Maschinenmodelle
  - Turing-Maschine
  - Registermaschine
  - Kellerautomat
  - Endlicher Automat

- Bis auf polynomielle Faktoren sind alle universellen Maschinenmodelle gleich mächtig → darum freie Wahl

- Gewähltes Maschinenmodell definiert Kosten für Arbeitsschritte

Kostenmaß

- Addition / Subtraktion
- Multiplikation / Division
- Vergleiche
- Etc.

➔ Abstrahieren auf uniformes Kostenmaß von 1

Eine Addition kostet 1

Kostenmaß

- Addition / Subtraktion
- Multiplikation / Division
- Vergleiche
- Etc.

➔ Abstrahieren auf uniformes Kostenmaß von 1

Eine Addition kostet 1

Kostenmaß

- Addition / Subtraktion
- Multiplikation / Division
- Vergleiche
- Etc.

➔ Abstrahieren auf uniformes Kostenmaß von 1

Eine Addition kostet 1

Typische Laufzeitanalysen

- Bester Fall
  Wie arbeitet der Algorithmus im günstigsten Fall?

- Durchschnittlicher Fall
  Wie arbeitet der Algorithmus im durchschnittlichen Fall?

- Schlechtester Fall
  Wie arbeitet der Algorithmus im schlechtesten Fall?

Typische Laufzeitanalysen

- Bester Fall
  Wie arbeitet der Algorithmus im günstigsten Fall?

- Durchschnittlicher Fall
  Wie arbeitet der Algorithmus im durchschnittlichen Fall?

- Armortisierter Worst-Case Fall
  Was sind die **durchschnittlichen** Kosten **einer** Operation in einer **schlechtest möglichen** Folge von Operationen

- Schlechtester Fall
  Wie arbeitet der Algorithmus im schlechtesten Fall?

Armortisierte Analyse

- Aggregatmethode

  Durchlauf „durchspielen“, Operationen zählen
  → Durchschnitt bilden

- Guthabenmethode / Bankkontomethode

  Mehr „bezahlen“, um spätere Kosten mitzutragen. Freie Wahl der Mehrkosten → Intuition gefrage

- Potentialmethode

  Mehr „bezahlen“ (siehe oben), jedoch über Funktion definiert

Tafelmitschrieb – Mitschreiben!

- Binärzählerbeispiel

Armortisierte Analyse

- Beispiel: selbst (schlecht) implementierte Pipe/Queue/FiFo

Java:

addElement(int a) → list.add(a)
getFirstElement() → tmp = list.get(0); list.remove(0); return tmp

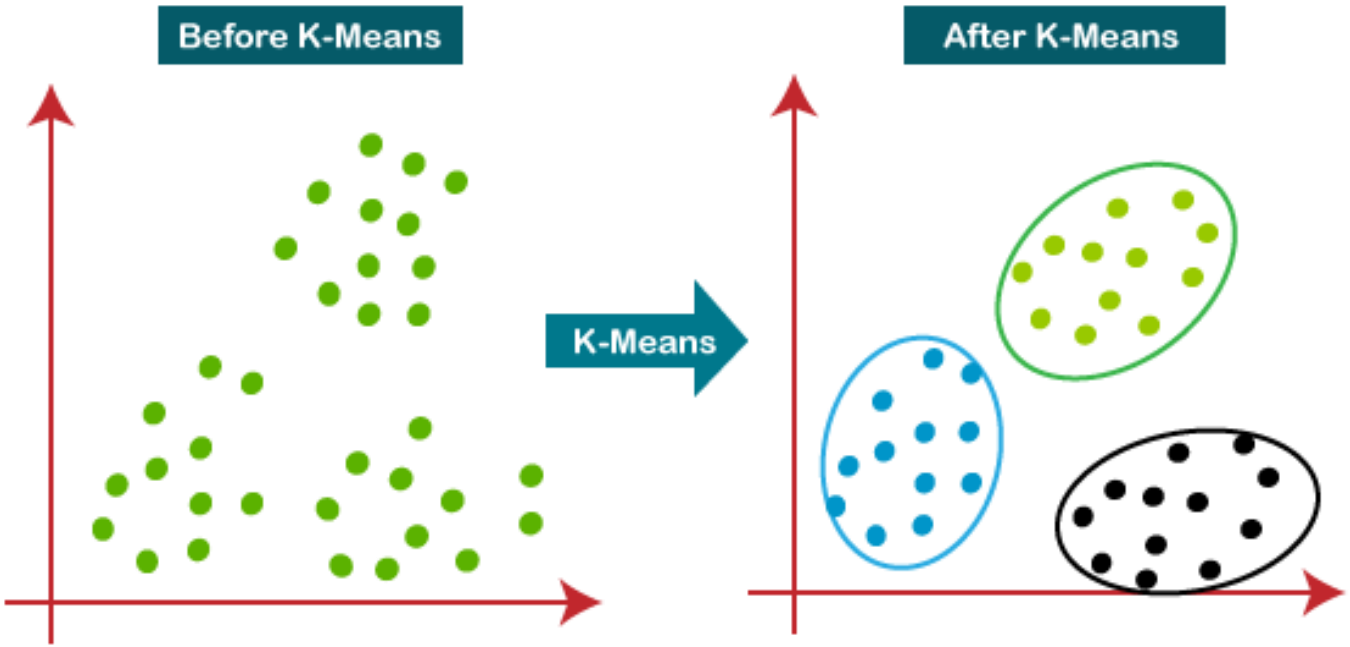
Analyse:
N Zahlen hinzufügen und irgendwann/zufällig wieder abrufen

Auswahl aus Big Data Analyse

- Cluster-Algorithmus

Clusteranalyse – K-Means

- Quelle Bilder: https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning (2021-09-14 letzter Aufruf)



K-Means Clustering

1. Choose the number of clusters(K) and obtain the data points
2. Place the centroids c_1, c_2, ... c_k randomly
3. Repeat steps 4 and 5 until convergence or until the end of a fixed number of iterations
4. for each data point x_i:
   - find the nearest centroid(c_1, c_2 .. c_k)
   - assign the point to that cluster
5. for each cluster j = 1..k
   - new centroid = mean of all points assigned to that cluster
6. End

K-Means Clustering – Notwendige Formeln

- Euklidische Distanz (https://de.wikipedia.org/wiki/Euklidischer_Abstand)

$$d(p,q) = \|q-p\|_2 = \sqrt{(q_1-p_1)^2+\cdots+(q_n-p_n)^2} = \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}$$

- Massezentrum (https://en.wikipedia.org/wiki/Center_of_mass)

$$\mathbf{R} = \frac{1}{M}\sum_{i=1}^{n} m_i \mathbf{r}_i,$$

- Notation Punkt:

$$p = (p_1, \ldots, p_n)$$

Aufgabe - Gruppen

- Cluster finden für gegebene Punkte:

z.B.: Alter der Studierenden und Größe in cm

z.B.: gemessene Geschwindigkeit, Alter FahrerIn, Alter Auto

etc.