Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. F. Kuhn
M. Fuchs, G. Schmid

Algorithmen und Datenstrukturen
Sommersemester 2026
Musterlösung Übungsblatt 2

Abgabe: Dienstag, 05. Mai, 2026, 10:00 Uhr

Aufgabe 1: Gehackt? *(10 Punkte)*

Ein Systemadministrator analysiert Login-Zeitpunkte, um verdächtiges Verhalten zu erkennen. Wenn ein User sich 'unmeschlich schnell' zweimal hintereinander eingeloggt hat, dann war es wahrscheinlich ein Bot. Gegeben ist eine Liste von n ganzen Zahlen, die Login-Zeitpunkte darstellen. Ziel ist es, die kleinste absolute Differenz zwischen zwei Zeitpunkten zu bestimmen.

Beispiel Gegeben sei folgendes Array:

$$A = [-20, 10, -18, 15, -100, 30, 98].$$

Gesucht ist die kleinste absolute Differenz zwischen zwei verschiedenen Elementen.
Formal bedeutet dies, dass wir den folgenden Wert bestimmen wollen:

$$\min_{1 \leq i < j \leq n} |A[i] - A[j]|.$$

Die kleinste dieser Differenzen im Array ist

$$|-20 - (-18)| = 2.$$

(a) Implementieren sie einen Brute-Force-Algorithmus, der alle Paare überprüft, um die minimale absolute Differenz zu bestimmen. Was ist die asymptotische Laufzeit von diesem Algorithmus? *(2 Punkte)*

(b) Entwerfen Sie einen effizienteren Algorithmus. Beschreiben Sie diesen zuerst in Worten und implementieren Sie ihn dann. Erläutern Sie, warum ist ihr Algorithmus korrekt und was ist die asymptotische Laufzeit von ihrem Algorithmus?[1] *(6 Punkte)*

(c) Auf der Webseite finden sie 'input.txt' mit 1000000 verschiedenen ganzen Zahlen. Verwenden Sie Ihre Implementierung, um das Ergebnis für diese Eingabe zu berechnen, und geben Sie das Resultat an. *(2 Punkte)*

Musterlösung

Die Musterlösung der Implementieraufgaben finden Sie auf der Website in der Datei 'analyze.py'.

[1] Wir haben euch auch eine 'compare_runtimes' Hilfsfunktion in der Vorlage bereitgestellt. Schaut euch gerne mal an wie eure bessere Version performed.

(a) Die BruteForce Lösung muss alle möglichen Paare an Indizes $i, j \in \{0, 1, 2, 3, \ldots, n-1\}$ testen. Selbst wenn die Implementierung jedes Paar nur einmal betrachtet (also nur (3,5) und nicht auch (5,3)), dann sind muss der erste Wert mit $n-1$ weiteren Werten gepaart werden, der zweite Wert mit $n-2$ und so weiter.

Für die insgesamte Anzahl an Paaren ergibt sich dann

$$n-1+n-2+n-3+\ldots+3+2+1 = \sum_{i=1}^{n-1} i = \frac{(n-1)n}{2} \in \Theta(n^2)$$

(b) Um das Problem effizienter zu lösen, können wir die Eingabe zuerst sortieren. In dem nun sortierten Array müssen die kleinste Differenz zwischen zwei benachbarten Zahlen sein. Dies folgt daraus, dass wenn $a < b < c$, dann $|a-b| < |a-c|$ für alle $a, b, c \in \mathbb{R}$.

Wenn wir das sortieren mit einem effizienten Sortieralgorithmus, wie z.B. MergeSort, implementieren, dann benötigt dieser Schritt asymptotisch $O(n \log n)$ Zeit.

Im zweiten Schritt, durchlaufen wir das sortierte Array um den Index i zu finden, so dass $A[i+1] - A[i]$ so klein wie möglich ist (weil $A[i+1] > A[i]$ ist benötigen wir keine Betragsklammern). Dafür reicht es einmal durch das Array zu iterieren, weswegen dieser Schritt in $O(n)$ Zeit ausgeführt werden kann.

Insgesamt ist die Laufzeit also $O(n \log n) + O(n) = O(n \log n)$.

Bei korrekter Implementierung ergibt sich im Schaubild von 'compare_runtimes' das gleiche Bild wie im letzten Übungsblatt (aus genau den gleichen Gründen).

(c) Der Korrekte Output für die Eingabe in 'input.txt' ist 619145. Der BruteForce Algorithmus sollte nicht effizient genug gewesen sein um dieses Problem zu lösen, unsere Implementierung des Ansatzes in Aufgabe (b) hingegen gibt die Antwort quasi sofort.

Aufgabe 2: O-Notation *(10 Punkte)*

(a) Beweisen oder widerlegen Sie unter Verwendung der Definition[2] der Landau-Notation aus der Vorlesung die folgenden Aussagen:

$$\sum_{i=0}^{n} i \in \Omega(n^2)$$

(2 Punkte)

$$n^{1/2} \cdot n^{1/3} \cdot n^{1/4} \in \Theta(n)$$

(2 Punkte)

$$3n \log n + 7n \in \Theta(n \log n)$$

(2 Punkte)

(b) Geben Sie zwei Funktionen f, g an, sodass gilt:

$$f(n) \in O(g(n)) \quad \text{aber} \quad f(n) \notin \Theta(g(n)).$$

Beweisen Sie, dass Ihre Funktionen diese Eigenschaften erfüllen. Sie dürfen hierfür Grenzwerte verwenden. *(4 Punkte)*

[2] Ein Beweis ausschließlich mit Grenzwerten führt zu 0 Punkten.

Musterlösung

(a) Zuerst formen wir um:

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n$$

Wir zeigen $\sum_{i=0}^{n} i \in \Omega(n^2)$.

Nach Definition von Ω müssen Konstanten $c > 0$ und $n_0 \in \mathbb{N}$ existieren, sodass für alle $n \geq n_0$ gilt:

$$\sum_{i=0}^{n} i \geq c \cdot n^2.$$

Es gilt:

$$\frac{1}{2}n^2 + \frac{1}{2}n \geq \frac{1}{2}n^2 \quad \text{für alle } n \geq 0.$$

Wähle $c = \frac{1}{2}$ und $n_0 = 0$. Damit ist die Aussage bewiesen.

Zuerst formen wir um:

$$n^{1/2} \cdot n^{1/3} \cdot n^{1/4} = n^{\frac{1}{2}+\frac{1}{3}+\frac{1}{4}} = n^{\frac{13}{12}}.$$

Da $\frac{13}{12} > 1$, wächst $n^{13/12}$ strikt schneller als n.

Für $\Theta(n)$ müsste gelten:

$$c_1 n \leq n^{13/12} \leq c_2 n.$$

Die rechte Ungleichung gilt nicht füt alle beliebig großen n, da:

$$n^{13/12} \leq c_2 n \Rightarrow \frac{n^{13/12}}{n} \leq c_2 \Rightarrow n^{1/12} \leq c_2$$

und egal wie die Konstante c_2 gewählt ist, für $n > c_2^{12}$ ist die Ungeleichung falsch.

Also gilt:

$$n^{1/2} \cdot n^{1/3} \cdot n^{1/4} \notin \Theta(n).$$

Wir zeigen:

$$3n \log n + 7n \in \Theta(n \log n).$$

Obere Schranke:

Für $n \geq 2$ gilt $\log n \geq 1$, also:

$$7n \leq 7n \log n.$$

Damit:

$$3n \log n + 7n \leq 3n \log n + 7n \log n = 10n \log n.$$

Also gilt mit $c_2 = 10$:

$$3n \log n + 7n \leq c_2 n \log n.$$

Untere Schranke:

Da $7n \geq 0$:

$$3n \log n + 7n \geq 3n \log n.$$

Also mit $c_1 = 3$:

$$3n \log n \leq 3n \log n + 7n.$$

Insgesamt:

$$3n \log n \leq 3n \log n + 7n \leq 10n \log n \quad \text{für } n \geq 2.$$

Damit ist die Aussage bewiesen.

(b) Wähle:

$$f(n) = n, \quad g(n) = n^2.$$

Zeige: $f(n) \in O(g(n))$

Es gilt:

$$n \leq n^2 \quad \text{für alle } n \geq 1.$$

Also mit $c = 1$, $n_0 = 1$:

$$f(n) \leq c \cdot g(n).$$

Damit $f(n) \in O(g(n))$.

Zeige: $f(n) \notin \Theta(g(n))$

Betrachte:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n}{n^2} = \lim_{n\to\infty} \frac{1}{n} = 0.$$

Damit existiert keine Konstante $c_1 > 0$, sodass gilt:

$$c_1 g(n) \leq f(n).$$

Also:

$$f(n) \notin \Omega(g(n)) \Rightarrow f(n) \notin \Theta(g(n)).$$