

Aufgabenblatt 2

Aufgabe 1, Vollständige Induktion (2P)

Zeigen Sie mittels Induktion, dass für alle Zweierpotenzen $n = 2^k$ und

$$T(n) = \begin{cases} 2 & \text{für } n = 2 \\ 2T(\frac{n}{2}) + n & \text{für } n = 2^k, k > 1 \end{cases}$$

gilt: $T(n) = n \cdot \log_2(n)$.

Aufgabe 2, Binary search (3P)

Gegeben sei eine geordnete Sequenz A , und eine Zahl v .

Schreiben Sie in Pseudocode einen Algorithmus, der bestimmt, ob v in A vorkommt. Beweisen Sie, dass die Laufzeit des Algorithmus höchstens von der Größenordnung $\Theta(\log_2(n))$ ist.

Aufgabe 3, Laufzeit verbessern (2P)

Beim insertion sort Algorithmus aus der Vorlesung werden in der while-Schleife alle Elemente von der Position $j - 1$ an rückwärts mit dem aktuellen Element verglichen. Kann der binary search algorithmus aus der vorhergehenden Aufgabe dazu verwendet werden, die Laufzeit des insertion sort auf die Größenordnung $\Theta(n \cdot \log_2(n))$ zu verbessern?

Aufgabe 4, Algorithmus (3P)

Gegeben sei eine Menge S von natürlichen Zahlen, und eine weitere natürliche Zahl x . Schreiben Sie einen Algorithmus (in Pseudocode), der in $\Theta(n \cdot \lg(n))$ Zeit bestimmt, ob es zwei Zahlen $a, b \in S$ gibt so dass $a + b = x$ gilt.