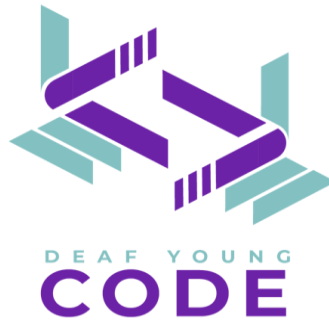




2023-2-IT03-KA220-YOU-000179130

Arbeitsblätter für Modul 4

Algorithmus

Verwendung als Einzelarbeit oder Teamarbeit



Co-funded by
the European Union

Finanziert durch die Europäische Union. Die geäußerten Ansichten und Meinungen sind jedoch ausschließlich die der Autoren und spiegeln nicht unbedingt die der Europäischen Union oder der Exekutivagentur Bildung, Audiovisuelles und Kultur (EACEA) wider. Weder die Europäische Union noch die EACEA können dafür verantwortlich gemacht werden.

Arbeitsblatt Algorithmus

Modul: Algorithmus

Dauer: 90 Minuten

Level: Anfänger*innen

Zielgruppe: Gehörlose Jugendliche, die Coding lernen

Teil 1: Warm-up (Basis)

1. Summe von 2 Ziffern

Berechne die Summe zweier einstelliger Zahlen.

Eingabe: Zwei einstellige Zahlen.

Ausgabe: Die Summe dieser Zahlen.

Beispiel: $2 + 3 = 5$

Umsetzung eines Algorithmus:

- Pseudocode
- C++
- Java
- Python

2. Maximales paarweises Produkt

Problem des maximalen paarweisen Produkts:

Finden das maximale Produkt zweier unterschiedlicher Zahlen in einer Folge von nicht-negativen ganzen Zahlen.

Eingabe: Eine Folge von nicht-negativen ganzen Zahlen.

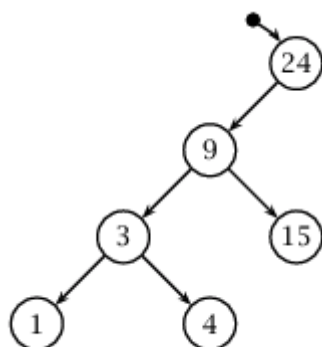
Ausgabe: Der maximale Wert, der durch Multiplikation zweier unterschiedlicher Elemente aus der Folge erhalten werden kann.

Beispiel:

	5	6	2	7	4
5		30	10	35	20
6	30		12	42	24
2	10	12		7	4
7	35	42	14		28
4	20	24	8	28	

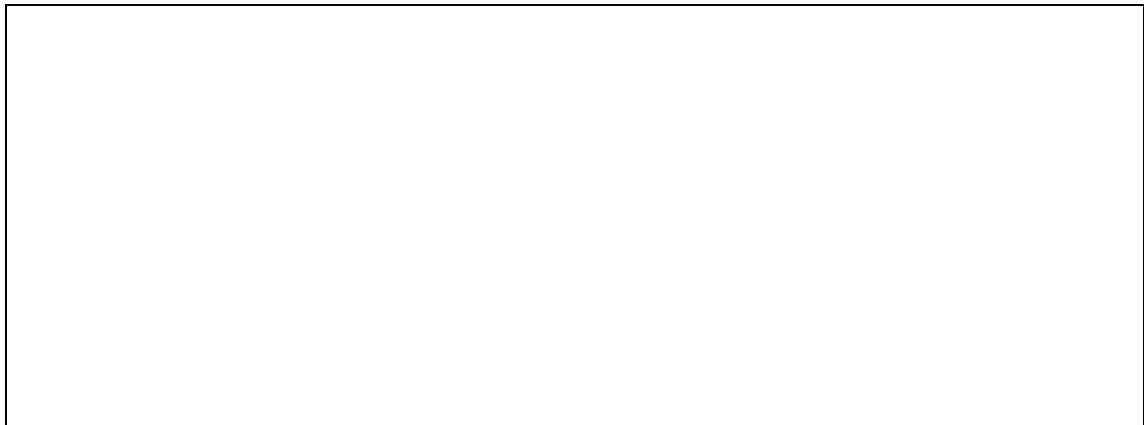
Teil 2 — Übung: "Binary search tree" (= binärer Suchbaum)

Schau dir den "Binary search tree" (BST) an:

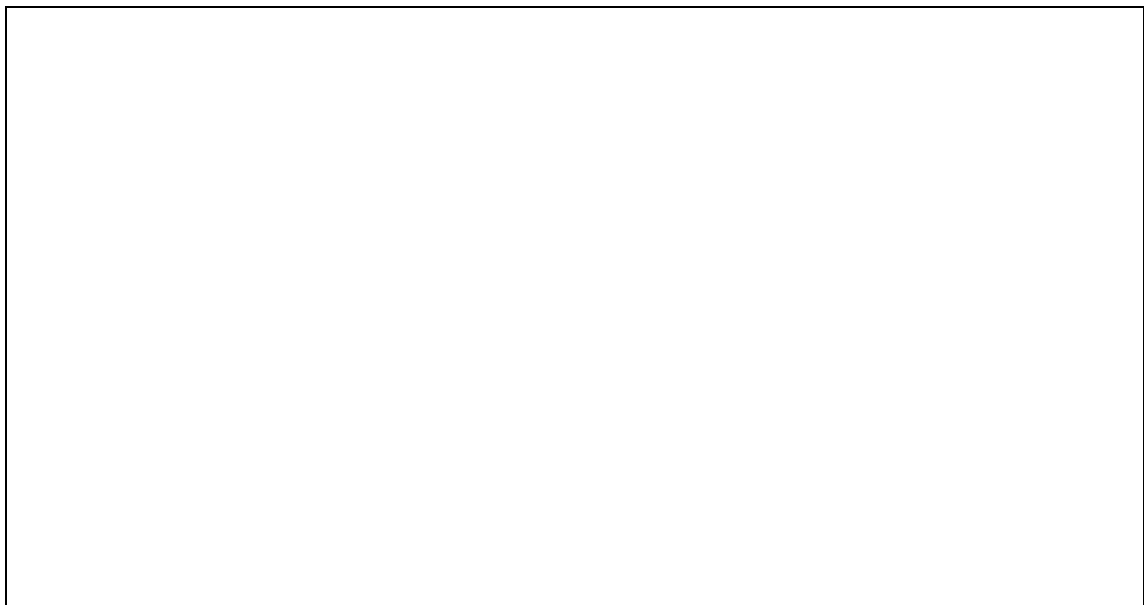


Frage 1: Liste alle möglichen Einfüge-Ordnungen (d. h. Permutationen) der Schlüssel auf, die diesen BST erzeugt haben könnten.

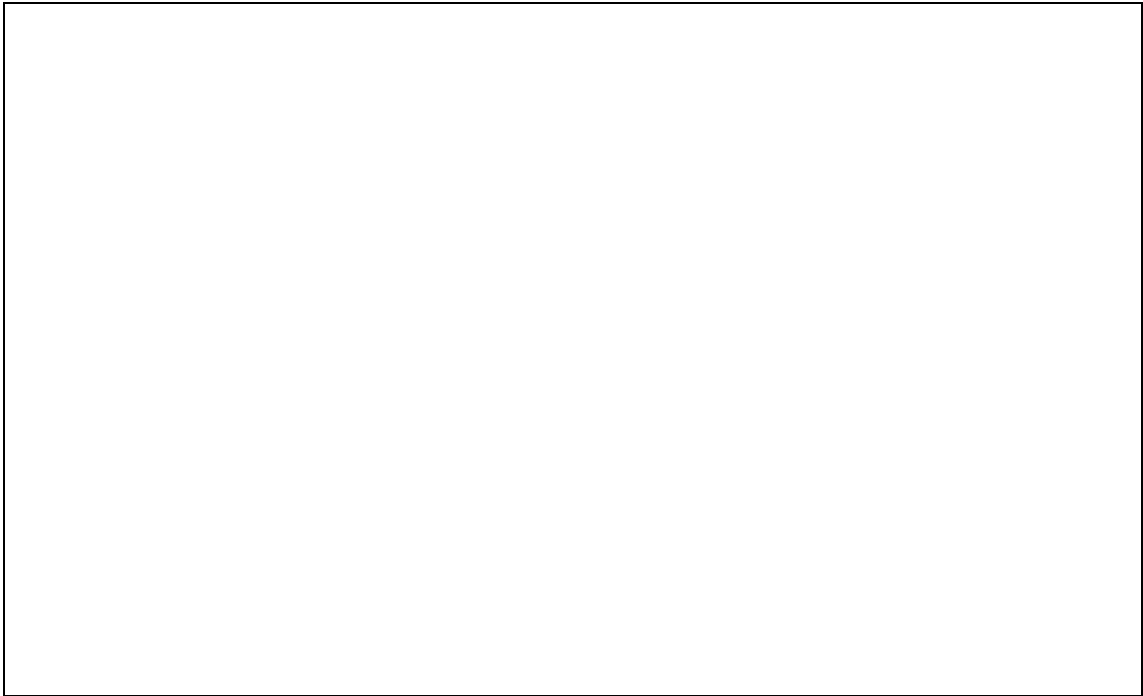
(Erklärung: Eine Permutation ist eine bestimmte Anordnung oder Vertauschung von Objekten in einer Reihenfolge.)

A large, empty rectangular box with a thin black border, intended for the student to list all possible insertion orders (permutations) of the keys that could have created the given BST.

Frage 2: Zeichne denselben BST nach dem Einfügen der Schlüssel 6, 45, 32, 98, 55 und 69 in dieser Reihenfolge.

A large, empty rectangular box with a thin black border, intended for the student to draw the resulting BST after inserting the keys 6, 45, 32, 98, 55, and 69 in that specific order.

Frage 3: Zeichne den BST, der sich aus der Löschung der Schlüssel 9 und 45 aus dem BST aus Frage 2 ergibt.



Frage 4: Schreibe mindestens drei Einfügeordnungen (Permutationen) der nach Frage 3 im BST verbleibenden Schlüssel auf, die einen ausgeglichenen Baum (d. h. einen Baum mit minimaler Höhe) ergeben würden.



Teil 3 – Übung: „Einfügungssortierung“

Schreibe den Pseudocode des Einfügungs-Sortier-Algorithmus und veranschauliche die Ausführung anhand der Reihe.

A = (7,17,89,74,21,7,43,9,26,10)

Schreibe dazu den Inhalt des Arrays bei jeder Hauptiteration (äußeren Iteration) des Algorithmus auf.

Schreibe dazu den Inhalt der Reihe bei jeder Hauptiteration (äußeren Iteration) des Algorithmus auf. Iteration = Wiederholung

Teil 4 — Übung: "max-heap"

Heap als Datenstruktur

Ein Heap ist ein spezieller binärer Baum, in dem an der Wurzel immer das **größte Element (Max-Heap)** oder das **kleinste Element (Min-Heap)** steht.

Schau dir den folgenden Max-Heap an:

H= 37,12,30,10,3,9,20,3,7,1,1,7,5

Write the exact output of the following Extract-All algorithm run on H

Extract-All(H)

```
1 while H.heap-size>0
2   Heap-Extract-Max(H)
3   for i=1 to H.heap-size
4     output H[i]
5   output "." end-of-line
```

Heap-Extract-Max(H)

```
1 if H.heap-size>0
2   k = H[1]
3   H[1] = H[H.heap-size]
4   H.heap-size = H.heap-size-1
5   Max-Heapify(H)
6 return k
```

Teil 5 — Quiz

Was ist ein Algorithmus?

- A. Eine Computermarke.
- B. Eine Schritt-für-Schritt-Anleitung zur Lösung eines Problems.
- C. Eine Art von Daten.
- D. Ein Passwort.

Welches dieser Beispiele ist ein alltägliches Beispiel für einen Algorithmus?

- A. Ein Kochrezept
- B. Ein Fußball
- C. Ein Musikvideo
- D. Ein Bild

Wie nennen wir die Informationen, die wir in einen Algorithmus eingeben?

- A. Ausgabe
- B. Eingabe
- C. Fehler
- D. Flussdiagramm

Was passiert, wenn ein Algorithmus einen Fehler (Bug) enthält?

- A. Er arbeitet schneller.
- B. Er stoppt oder liefert falsche Ergebnisse.
- C. Er wird sicherer.
- D. Er verschwindet.

Was ist die richtige Reihenfolge eines Algorithmus?

- A. Ausgabe → Prozess → Eingabe
- B. Prozess → Eingabe → Ausgabe
- C. Eingabe → Prozess → Ausgabe

Teil 6 — Reflexion, Feedback

Beantworte die Fragen mit kurzen Sätzen oder Ausdrücken.

Was hast du heute über Algorithmen gelernt?

Was war schwierig für dich?