Aufgabe 1: Drehfreudig

Team-ID: 00572

Team-Name: Ralli

Bearbeiter/-innen dieser Aufgabe:
David Adam

20. Oktober 2025

Inhaltsverzeichnis

1 Lösungsidee

Die Kernidee meiner Lösung basiert darauf, dass ein Baum genau dann drehfreudig ist, wenn die Blattrechtecke unter 180°-Drehung auf sich selbst abgebildet werden. Jeder Knoten im Baum bekommt ein horizontales Intervall zugewiesen, das seinen Platzbereich im Gesamtrechteck definiert.

Ich starte bei der Wurzel mit dem Intervall [0, 1] und teile dieses gleichmäßig auf alle Kindknoten auf. Wenn ein Knoten zum Beispiel 3 Kinder hat, bekommt jedes Kind genau ein Drittel der Breite des Elternintervalls. Diese Aufteilung mache ich rekursiv für den ganzen Baum durch, bis ich bei den Blättern ankomme.

Die Drehfreudigkeit überprüfe ich dann so: Ich schaue mir alle Blattintervalle an und drehe sie gedanklich um 180° im Einheitsrechteck. Das heißt, ein Intervall [s, s+w) wird zu [1-(s+w), 1-s). Wenn die Menge aller gedrehten Intervalle exakt mit der ursprünglichen Menge übereinstimmt, ist der Baum drehfreudig.

Für die Visualisierung generiere ich eine SVG-Grafik, die den Baum zweimal zeigt: einmal normal und einmal um 180° gedreht. Dadurch kann man direkt sehen, ob die Blätter wieder übereinander liegen.

2 Umsetzung

Die Umsetzung folgt der beschriebenen Idee. Zuerst verarbeite ich die Klammer-Eingabe und erzeuge mit einem Stack die Kindliste `children`. Anschließend weist `compute_intervals` per Tiefensuche jedem Knoten ein exaktes Intervall zu (mit `Fraction`, um Rundungsfehler zu vermeiden), indem die aktuelle Breite gleichmäßig auf die Kinder verteilt wird. Die Drehfreudigkeit prüfe ich, indem ich alle Blattintervalle zu `(1-(s+w), w)` spiegele und die sortierten Listen vergleiche. Falls die Bedingung erfüllt ist, erstellt `generate_svg` eine Bild: oben der Originalbaum, unten die 180° gedrehte Variante, getrennt durch eine Mittellinie.

3 Werkzeuge

Für diese Aufgabe habe ich folgende Werkzeuge verwendet:

fractions.Fraction: Diese Python-Bibliothek nutze ich für exakte Bruchrechnung. Damit vermeide ich Rundungsfehler beim Vergleich der Intervalle. Die Intervallbreiten sind oft Brüche wie 1/3 oder 1/4, und mit Fraction bleiben diese Werte exakt.

ChatGPT-5: Ich habe ChatGPT genutzt, um mir erklären zu lassen, wie man SVG-Grafiken erstellt. Besonders die Syntax für Linien, Kreise und das viewBox-Attribut waren für mich neu. ChatGPT hat mir Beispielcode gezeigt, den ich dann an meine Bedürfnisse angepasst habe.

VS Code mit Inline-Vervollständigung: Beim Schreiben des Codes hat mir die Inline-Vervollständigung von VS Code (GitHub Copilot) geholfen, vor allem bei sich wiederholenden Konstrukten wie den SVG-Tags und bei der Formatierung der Ausgabe. Die Vorschläge waren meistens sinnvoll und haben mir Zeit gespart.

Vorgehensweise: Zuerst habe ich mir die Aufgabenstellung genau durchgelesen und ein paar Beispiele auf Papier durchgerechnet, um die Idee mit den Intervallen zu verstehen. Dann habe ich angefangen, das Parsen zu programmieren und mit kleinen Testbäumen überprüft. Als das funktionierte, kam die Intervallberechnung dran. Die SVG-Generierung war der letzte Schritt, weil ich dafür erst recherchieren musste, wie SVG überhaupt funktioniert.

4 Beispiele

4.1 drehfreudig01.txt

```
Der Baum ist drehfreudig.
```

4.2 drehfreudig02.txt

```
Der Baum ist nicht drehfreudig.
```

4.3 drehfreudig03.txt

```
Der Baum ist nicht drehfreudig.
```

4.4 drehfreudig04.txt

```
Der Baum ist drehfreudig.
```

4.5 drehfreudig05.txt

```
Der Baum ist nicht drehfreudig.
```

4.6 drehfreudig06.txt

```
Der Baum ist drehfreudig.
```

4.7 drehfreudig07.txt

```
Der Baum ist nicht drehfreudig.
```

4.8 drehfreudig08.txt

```
Der Baum ist drehfreudig.
```

4.9 drehfreudig09.txt

```
Der Baum ist nicht drehfreudig.
```

4.10 drehfreudig10.txt

```
Der Baum ist drehfreudig.
```

4.11 drehfreudig11.txt

```
Der Baum ist nicht drehfreudig.
```

4.12 drehfreudig12.txt

```
Der Baum ist nicht drehfreudig.
```

4.13 drehfreudig13.txt

```
Der Baum ist drehfreudig.
```

4.14 drehfreudig14.txt

```
Der Baum ist nicht drehfreudig.
```

4.15 drehfreudig15.txt

```
Der Baum ist drehfreudig.
```

5 Quellcode (Auszug)

```python
def compute_intervals(root, children):
    """Berechnet␣fuer␣jeden␣Knoten␣sein␣horizontales␣Intervall."""
    intervals = {}
    max_depth = 0

    def dfs(u, start, width, depth):
        nonlocal max_depth
        intervals[u] = (start, width, depth)
        max_depth = max(max_depth, depth)
        k = len(children[u])
        if k:
            child_width = width / k
            x = start
            for v in children[u]:
                dfs(v, x, child_width, depth + 1)
                x += child_width

    dfs(root, Fraction(0, 1), Fraction(1, 1), 0)
    return intervals, max_depth


def is_drehfreudig(children, intervals) -> bool:
    """Prueft␣ob␣die␣Blatt-Rechtecke␣unter␣180-Drehung
␣␣␣␣␣␣␣zusammenpassen."""
    leaves = [u for u in range(len(children))
              if not children[u]]
    segs = sorted((intervals[u][0], intervals[u][1])
                  for u in leaves)
    mirrored = sorted(
        (Fraction(1, 1) - (s + w), w)
        for (s, w) in segs
```

```
    )
    return segs == mirrored
```