



Institut für Informationsverarbeitung (TNT)
WS 2022/2023



Appelstraße 9a, 30167 Hannover
www.tnt.uni-hannover.de

TNT Labor

Übung: Vorprüfung

Die Labore erfordern grundlegende Kenntnisse in Python und git. Implementieren Sie die folgenden Aufgaben und reichen Sie die Lösung wie unten beschrieben **bis Sonntag, den 9. Oktober 2022, 13:59 Uhr** ein.

Installation

Voraussetzungen

- Python 3.8
- Git

Klonen Sie das Repository

```
1 git clone git@github.com:tntlabor2022/vorpruefung-USERNAME.git
2 cd Vorpruefung-USERNAME
```

wobei die URL sowie der Ordnername von Ihnen angepasst werden muss.

Installieren Sie die Abhängigkeiten

```
1 pip install -r requirements.txt
```

Stellen Sie sicher, dass `pytest` installiert ist, indem Sie das Programm über die Kommandozeile aufrufen. Sollte dies nicht der Fall sein, folgen Sie der Installationsanleitung <https://docs.pytest.org/en/stable/getting-started.html>.

Es dürfen nur die in `requirements.txt` aufgeführten Pakete verwendet werden.

Test

Sie können Ihre Implementierung mit `pytest` testen. Führen Sie dazu das folgende Kommando aus:

```
1  pytest
```

Die jeweiligen Aufgaben können auch einzeln getestet werden:

```
1  pytest test_1_trees.py
```

Beachten Sie, dass die Ihnen gegebenen Testfälle dazu dienen, Ihnen die Überprüfung der Korrektheit zu erleichtern. Das Bestehen der gegebenen Testfälle bedeutet nicht, dass Ihre Lösung korrekt ist. Insbesondere behalten wir uns vor, Ihre Lösung mit weiteren Testfällen automatisiert zu testen, um zu verhindern, dass Soll-Ausgaben fest einprogrammiert werden. Zusätzliche Testfälle haben niemals eine höhere Komplexität als die Ihnen gegebenen.

Aufgaben

Implementieren Sie die folgenden zwei Aufgaben.

Aufgabe 1: Kombination von Bäumen

In dieser Aufgabe erhalten Sie eine variable Anzahl von gewurzelten Bäumen, deren Kanten über Namen und deren Blattknoten über Zahlenwerte verfügen. Innere Knoten haben keinen Wert. Der Pfad von der Wurzel zu jedem Blattknoten ist eindeutig über die Sequenz der Kantennamen bestimmt. Alle Bäume pro Aufruf haben die gleiche Menge aller Pfade (sie unterscheiden sich höchstens in den Blattknotenwerten). Fügen Sie die Bäume zu einem einzigen Baum zusammen. Dabei soll für jedes Blatt der neue Wert durch Konkatenierung der entsprechenden Blattknotenwerte der Eingabebäume als Tupel angelegt werden. Behalten Sie bei der Konkatenierung die Reihenfolge der Eingabe bei. Ein Beispiel finden Sie in Abbildung 1.

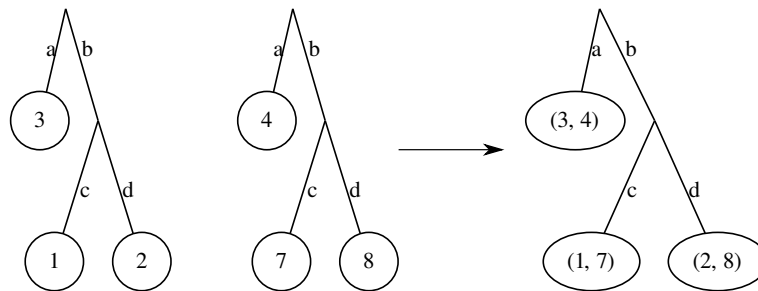


Abbildung 1: Beispielhafte Kombination der beiden linken Bäume in den rechten Baum.

Innere Knoten des Baumes werden dabei als `dict` kodiert. Die Schlüssel stellen die Namen der Ausgangskanten dar. Der Wert unter dem Schlüssel ist die Kodierung des auf diese Kante folgenden Knotens. Blattknoten werden durch den Wert des Blattes kodiert, wobei dieser einen beliebigen Typ außer `dict` haben darf. Der gesamte Baum wird durch den Wurzelknoten ausgedrückt. Der erste Baum aus Abbildung 1 wird wie folgt kodiert:

```
1  tree = {'a': 3, 'b': {'c': 1, 'd': 2}}
```

Implementieren Sie die Funktion in `exercise_1_trees.py`. Ihre Lösung muss ausschließlich in dieser Datei enthalten sein.

Aufgabe 2: Erkennung auf Binärbild

Gegeben ist ein binäres Bild mit Breite w und Höhe h in dem ein weißes Quadrat auf schwarzem Hintergrund dargestellt ist, wie in Abbildung 2. Bestimmen Sie den Mittelpunkt $(x, y)^T$, die Kantenlänge l , sowie den Drehwinkel α des Quadrates. Implementieren Sie Ihre Lösung in `exercise_2_rect.py`. Ihre Lösung muss ausschließlich in dieser Datei enthalten sein. Beachten Sie:

- Das Bild liegt in der Form (h, w) vor.
- Das Quadrat ist vollständig im Bild enthalten.
- Der Koordinatenursprung liegt oben rechts, die x -Achse zeigt nach rechts, die y -Achse nach unten.
- Der geforderte Winkel α ist in Grad anzugeben, die positive Rotationsrichtung ist gegen den Uhrzeigersinn, 0° zeigt nach oben. Auf Grund der Symmetrie ist jeder Winkel, der durch Rotationen um 90° auf den Lösungswinkel gebracht werden kann, akzeptabel.
- Für die Lösung der Aufgabe ist kein Vorwissen über Morphologie, Bilderkennung, etc. erforderlich. Nutzen Sie keine Funktionen der `scipy`-Bibliothek.

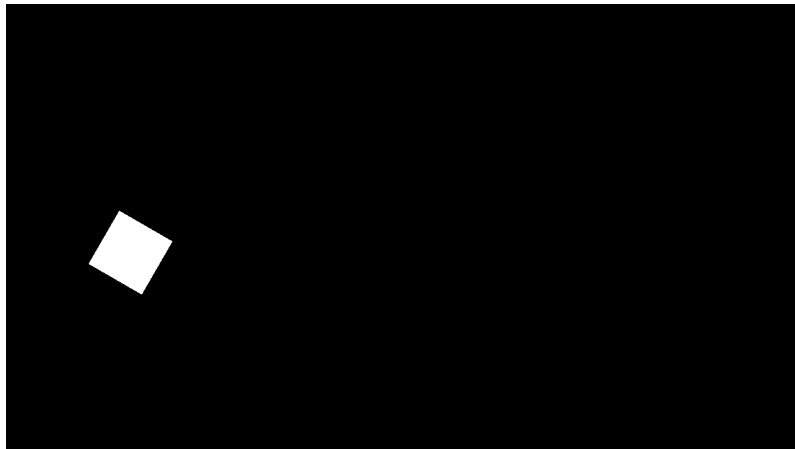


Abbildung 2: Beispielhaftes Eingabebild

Abgabe

Implementieren Sie die Aufgaben wie in der Aufgabenstellung beschrieben. Die Tests dienen als Hilfestellung für Ihre Implementierung. Die Test-Dateien dürfen nicht geändert werden. Die Abgabe soll auf dem *main* Branch erfolgen.

Wenn Sie alle Aufgaben implementiert haben und die Tests ohne Fehler durchlaufen, committen und pushen Sie den Code zu Github. Anschließend geben Sie Ihre Lösung über das folgende Formular ab:

<https://survey.uni-hannover.de/index.php/824449>

Wir können nur vollständige und funktionsfähige Implementierungen berücksichtigen. Wir akzeptieren weiterhin nur Lösungen, welche eine angemessene Effizienz aufweisen. Lösungen, welche für die Gesamtheit der Ihnen gegebenen Testfälle auf einem aktuellem Computer des Institutes mehr als 60 Sekunden benötigen, werden nicht akzeptiert. Dies ist das 200-fache der Zeit, die eine nicht optimierte Musterlösung benötigt.

Bei technischen Problem schreiben Sie bitte eine E-Mail an *betreuer_kilabor@tnt.uni-hannover.de*.