Aufgabe 2.1 (6 Punkte)
Sortieren Sie folgende Schlüssel mit Hilfe von Lexicographic Sort:

`REISE`, `MESSE`, `REGIE`, `ESSIG`, `IMMER`, `EIMER`, `GREIS`, `SERIE`, `EMSIG`.

Geben Sie den Inhalt von `QUEUE` nach jedem Durchlauf der äußeren Schleife an.

Aufgabe 2.2 (6 Punkte)
Die *Tiefe eines Knotens* in einem Binärbaum ist die Länge des Pfades von der Wurzel zum Knoten. Ein Binärbaum heißt *vollständig* genau dann, wenn es ein $k \in \mathbb{N}_0$ gibt, so dass jeder Knoten mit einer Tiefe kleiner als k zwei Kinder besitzt, und jeder Knoten mit der Tiefe k ein Blatt ist.

Implementieren Sie eine rekursive Prozedur, die für einen durch die Arrays `Leftchild` und `Rightchild` gegebenen Binärbaum prüft, ob dieser vollständig ist.

Aufgabe 2.3 (6 Punkte)
Gegeben sei die rekursive Funktion

$$k(n) = \frac{2}{n^2} \sum_{i=0}^{n-1} i \cdot k(i) + 1\,.$$

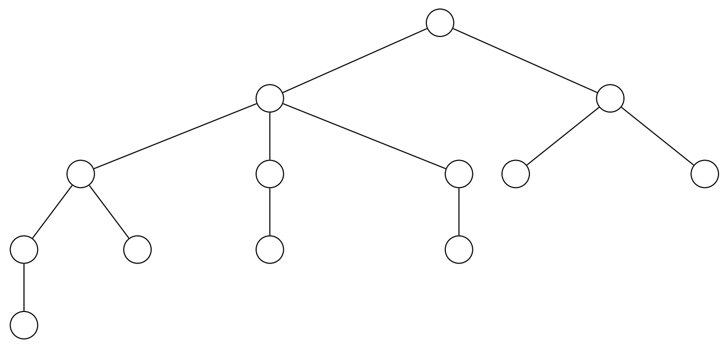
Implementieren Sie eine Prozedur `AvgPathLen(n)` zur Berechnung der Funktion $k(n)$ mit einem Zeitbedarf von $O(n)$.

<u>Hinweis:</u> Man kann zeigen, dass die Funktion $k(n)$ die durchschnittliche Tiefe eines beliebigen Knotens in einem beliebigen Binärbaum mit n Knoten berechnet.

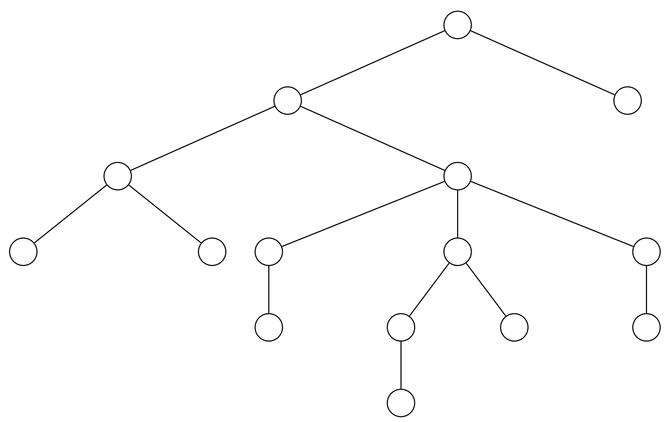
Bitte wenden!

Aufgabe 2.4 (6 Punkte)
Untersuchen Sie die beiden folgenden Bäume T_1 und T_2 unter Verwendung des Algorithmus aus der Vorlesung auf Isomorphie:

T_1:



T_2:



Geben Sie zu jedem inneren Knoten in T_1 und T_2 das zugehörige berechnete k-Tupel an, wobei k die Anzahl seiner Kinder sei.

Abgabe: Lösungen können jeweils bis zum folgenden Dienstag um 12:00 Uhr in die Kästen vor NA 02/257 (Nähe Rechenzentrum Servicecenter) *nach Aufgaben getrennt* eingeworfen werden. Geben Sie Ihren Namen, Ihre Matrikelnummer und Ihre Gruppe an. Auf jedem abgegebenen Aufgabenzettel dürfen bis zu drei Namen stehen.