Aufgabe 1

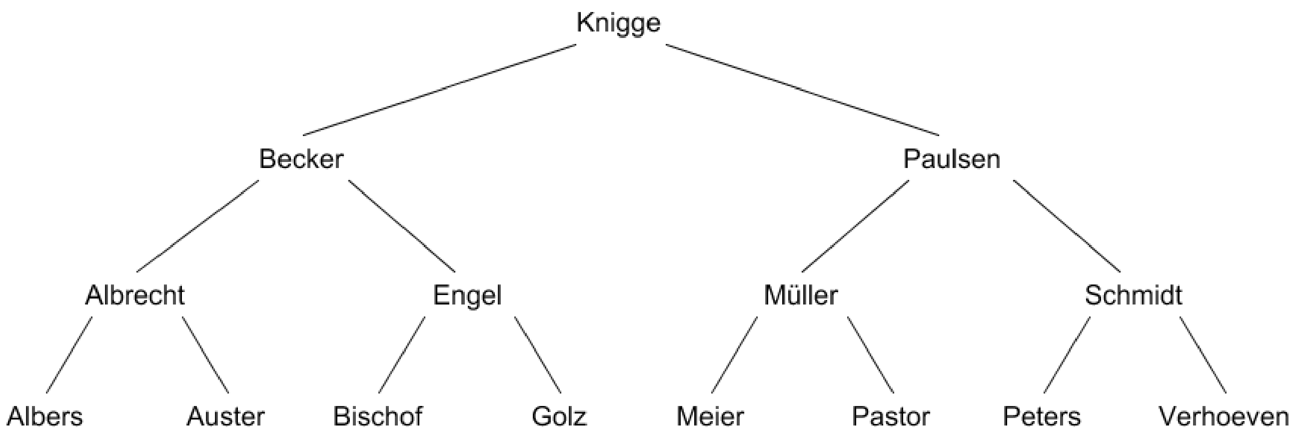
Falls der erste Buchstabe des 1. Wortes alphabetisch vor dem ersten Buchstaben des 2. Wortes liegt, liegt das 1. Wort alphabetisch vor dem 2. Wort.
Falls die ersten Buchstaben gleich sind, vergleiche die zweiten Buchstaben der Wörter auf die gleiche Weise, dann ggf. die dritten, usw.

Sonderfall: Falls alle Buchstaben der Wörter gleich sind, aber das 2. Wort zusätzliche Buchstaben enthält, ist das 1. Wort alphabetisch vor dem 2. Wort.
Bsp.: neu < neuer

Es gibt mehrere Möglichkeiten, Umlaute einzusortieren
Mgl. 1: ua … ue … ü … uf … uz (diese Regel wenden wir in den Aufgaben an)
Mgl. 2: a … b … c … z … ä … ö … ü
Mgl. 3: ua … uz … ü

Aufgabe 2

Hier schleichen sich schnell kleine Fehler ein, diese Aufgabe sollte daher mit Geduld bearbeitet werden. Überprüfen kann man anschließend anhand des Kriteriums:
Alle Knoten des linken TB müssen vor dem Elternknoten kommen, alle Kn. des rechten nachher.



Aufgabe 3

Berechnung des Mittelwert: Berechne, wie viele Schritte es für jeden einzelnen Namen braucht und bilde die Summe. Dann Teile durch die Anzahl der Namen.

a) **Liste**: $(1 + 2 + \ldots + 15) / 15 = ((15 \cdot 16) / 2) / 15 = 8$
Zur Erinnerung: $1 + 2 + \ldots + n = n \cdot (n+1) / 2$

b) **Binärbaum**: $(1 + 2 + 2 + 3 + 3 + 3 + 3 + \ldots) / 15 = (1 \cdot 1 + 2 \cdot 2 + 4 \cdot 3 + 8 \cdot 4) / 15 = 3{,}27$

c) Man muss die ganze Liste durchlaufen, benötigt also 15 Schritte.
Im Binärbaum geht bis zu einem Blatt. In einem regelmäßig aufgebauten Baum (alle Äste sind vollständig gefüllt) mit 15 Namen haben alle Blätter eine Tiefe von 4, es braucht also 4 Schritte.

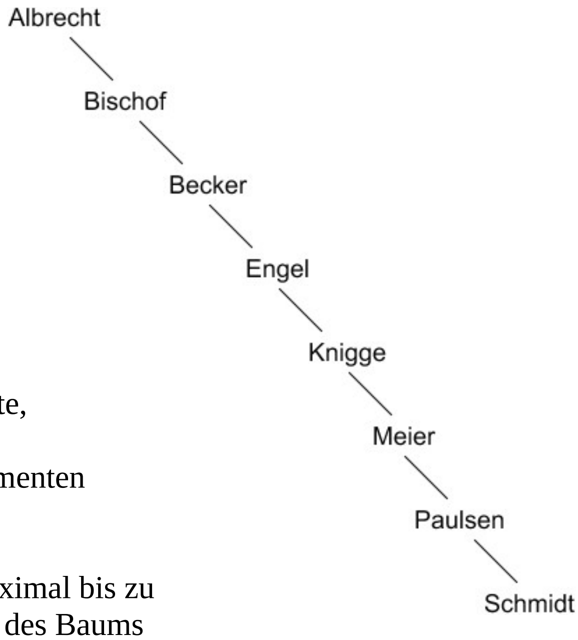
Zusatz für ca. 1000 Elemente:
Am günstigsten rechnen lässt sich mit $2^n - 1$ Elementen, also mit $2^{10} - 1 = 1023$

Liste: im Mittel 512 Schritte
$(1 + \ldots + 1023) / 1023 = ((1023 \cdot 1024) / 2) / 1023 = 1024 / 2 = 512$

Suchbaum: im Mittel 9 Schritte
$(1 \cdot 1 + 2 \cdot 2 + 3 \cdot 4 + 4 \cdot 8 + 5 \cdot 16 + 6 \cdot 32 + 7 \cdot 64 + 8 \cdot 128 + 9 \cdot 256 + 10 \cdot 512) / 1023 = 9{,}01$

Aufgabe 4

Die Knoten werden alle auf der rechten Seite des untersten Knoten eingefügt. Dadurch ergibt sich die Struktur einer Liste – und damit geht der Vorteil des Suchbaums gegenüber der Liste verloren.



Aufgabe 5

In der Liste benötigt die Suche maximal so viele Schritte, wie Elementen in der Liste sind.
Bei 1000 Elementen also 1000 Schritte, bei 1 Mio. Elementen 1 Mio. Schritte, bei n Elementen n Schritte.

Im regelmäßig gefüllten Suchbaum muss die Suche maximal bis zu den Blättern gehen. Man muss also die maximale Tiefe des Baums berechnen.

Ein Binärbaum mit maximaler Tiefe 2 hat 1 + 2 = 3 Knoten. Das entspricht $2^2 - 1$.
Mit Tiefe 3 kommen 4 Knoten hinzu, also 1 + 2 + 4 = 7 Knoten, oder $2^3 - 1$.
Mit Tiefe 4 weitere 8 Knoten, also 1 + 2 + 4 + 8 = 15 Knoten, oder $2^4 - 1$.
Mit Tiefe k hat ein Baum mit Tiefe n hat $2^k - 1$, oder rund 2^k Knoten.

Im Umkehrschluss heißt das: ein Baum mit n Knoten hat eine maximale Tiefe von $\log_2(n)$ Knoten.

Für n = 1000 Knoten: $\log_2(1000)$ ist ungefähr 10, weil $2^{10} = 1024$
Die Suche braucht also maximal 10 Schritte.

Für n = 1 Mio. Knoten ist $\log_2$(1 Mio.) ungefähr 20, weil $2^{20} = 2^{10} \cdot 2^{10} = 1024 \cdot 1024$ = rund 1 Mio.
Die Suche braucht also maximal 20 Schritte.