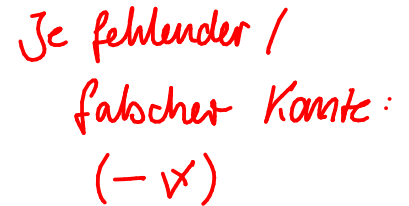


Aufgabe 1a



Ein ungerichteter Graph führt zu einer symmetrischen Adjazenzmatrix. Hier ist sie jedoch nicht symmetrisch, daher muss es sich um einen gerichteten Graphen handeln.

Der Graph ist zyklusfrei. Die Gewichtungen ergeben die Richtung der untergeordneten Elemente. z.B. 1 $\hat{=}$ links; 2 $\hat{=}$ rechts.

1d

```
graph TD; F((F)) --- C((C)); F --- I((I)); C --- B((B)); C --- D((D)); B --- A((A)); D --- E((E)); I --- H((H)); I --- J((J)); H --- G((G)); J --- K((K));
```

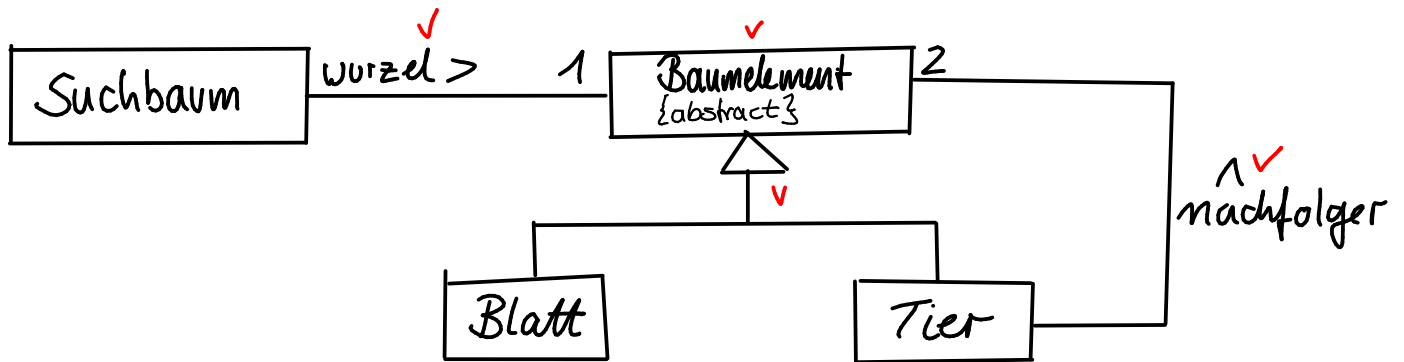
Wurzel

Rechts

Ganz

- ✓ Richtungen
- ✓ Gesamtstruktur

Aufgabe 2a



Aufgabe 2b

Durch einen Inorder-Durchlauf lässt sich der Baum zu einer sortierten Liste umwandeln. Man beginnt einen neuen Baum mit dem mittleren Element als Wurzel und teilt damit die Liste in zwei gleich große (± 1) Teile. Rekursiv wird nun immer das mittlere Element der verbliebenen Listen mit dem üblichen Algorithmus in den Baum eingefügt und damit wieder jede Teilliste in kleinere Listen gespalten bis diese elementar sind. Diese Elemente können nun zuletzt einsortiert werden.

Aufgabe 3a

Im genannten Dateisystem lassen sich nur relativ wenige Dateien (255)^{vv} in einem Ordner ablegen.

(Auch die maximale Ordneranzahl ist auf die gleiche Weise eingeschränkt) Anm. eine Einschränkung ist hinreichend

Aufgabe 3b

$d \triangleq$ Anzahl der Dateien (maximal)

$\sigma \triangleq$ Anzahl der Ordner (maximal) (inkl. Hauptordner)

$e \triangleq$ Anzahl der Ordnersebenen (inkl. Hauptordner)

$$\rightarrow e_1 = 1 \Rightarrow \sigma_1 = 1 \Rightarrow d_1 = 255$$

$$e_2 = 2 \Rightarrow \sigma_2 = 1 + 255 = 256 \Rightarrow d_2 = 256 \cdot 255 = 65280$$

$$e_3 = 3 \Rightarrow \sigma_3 = 256 + 255^2 = 65281 \quad \left. \vphantom{\begin{matrix} e_3 = 3 \\ \sigma_3 = 256 + 255^2 \\ = 65281 \end{matrix}} \right\} \Rightarrow d_3 = 65281 \cdot 255 = 16646655$$

$$e_4 = 4 \Rightarrow \sigma_4 = 65281 + 255^3 = 16646656 \quad \left. \vphantom{\begin{matrix} e_4 = 4 \\ \sigma_4 = 65281 + 255^3 \\ = 16646656 \end{matrix}} \right\} \Rightarrow d_4 = 16646656 \cdot 255 = 4244897280$$

$$d_3 < 19628327 < d_4$$

Es sind zumindest 4 Ebenen nötig, 3 reichen nicht aus.

Anm.: maximal 3 BE auf richtigen / sinnvollen Ansatz

„Einfacher“ / (falscher) Ansatz: $\sigma = 255^e$ o.Ä. \rightarrow 2 BE
maximal 2 BE auf Ergebnis

Aufgabe 3c

Die angegebene Methode gibt die Dateinamen aller
im Ordner liegender Dateien aus.

(vgl. „dir“ in DOS/Windows oder „ls“ in unixoiden OS)

Aufgabe 3d

In Festplatte

Anm:

1 BE

"return" statt
println eben-
falls i.O.

```
void suchen (String dn) {  
    String result = hauptordner.rsuchen(dn);  
    if (result.equals("")) {  
        System.out.println("Keine Datei gefunden");  
    } else {  
        System.out.println(result);  
    }  
}
```

In Ordner:

Anm:

Suche im Ordner:

max. 2 BE

```
String rsuchen (String dn) {  
    for (int i = 0; i < 255; i++) {  
        if (dateien[i] != 0 && dateien[i].Dateiname.equals(dn)) {  
            return Ordnername + dn;  
        }  
    }  
}
```

rekursive Suche
in allen Unter-
ordnern:

max. 4 BE

```
for (int i = 0; i < 255; i++) {  
    if (unterordner[i] != null) {  
        if (!unterordner[i].rsuchen(dn).equals("")) {  
            return Ordnername + unterordner[i].rsuchen(dn);  
        }  
    }  
}  
return "";  
}
```

Aufgabe 4

