

1. Du kannst drei Anwendungsgebiete von Trees angeben (Theorie S. 1–2).
2. Du kannst die passenden Fachausdrücke für Elemente eines gezeichneten Baums angeben: *Knoten*, *Kante*, *Wurzel*, *Elternknoten*, *Kindknoten*, *Geschwisterknoten*, *Blatt*, *innerer Knoten*, *Teilbaum* (Theorie S. 3)
3. Du verstehst, was ein *Pfad* ist und kannst für einen Knoten in einem gezeichneten Baum angeben, welche *Tiefe* und welche *Höhe* der Knoten hat. Ferner kannst du die *Höhe des Baums* angeben (Theorie S. 3).
4. Du kannst einen *Wald* erkennen (Theorie S. 3).
5. Du kannst die folgenden Typen von Binärbäumen erkennen: *entarteter Binärbaum*, *voller Binärbaum*, *balancierter Binärbaum*, *vollständiger Binärbaum*, *perfekter Binärbaum* (Theorie S. 3–4)  
zur Erinnerung: *Binärbäume* sind Bäume, deren Knoten maximal zwei Kindknoten haben.
6. Du kannst den Binärbaum skizzieren, der zu einer nicht allzu grossen Liste aus verschachtelten Listen besteht. (Theorie S. 5 oben)  
Die Syntax wird in der Form [Knotenwert, linker Teilbaum, rechter Teilbaum] angegeben.
7. Du kannst einen *Heap* (eine Liste ohne „nulltes“ Element, die einen vollständigen Binärbaum codiert) als Baum skizzieren und umgekehrt einen vollständigen Binärbaum als Heap darstellen.
8. Du kannst in einem Heap aus dem Index  $i$  eines Kindknotens den Index des zugehörigen Elternknotens berechnen ( $\lfloor i/2 \rfloor$ ) sowie aus dem Index eines Elternknotens  $j$  den Index des linken ( $2j$ ) und (sofern vorhanden) des rechten Kindknotens ( $2j + 1$ ) berechnen.
9. Du kannst erkennen, ob es sich bei einem Heap um einen Min-Heap, einen Max-Heap oder um keines von beiden handelt.
10. Du kannst an der graphischen Darstellung in einem Min-Heap zeigen, . . .
  - wie ein neues Element hinzugefügt und die Min-Heap-Eigenschaft mit „swim“-Operationen wieder hergestellt wird.
  - wie das kleinste Element entfernt und die Min-Heap-Eigenschaft mit „sink“-Operationen wieder hergestellt wird.

Es wird kein Python-Code zu Heaps geprüft und es muss auch kein (ungeordneter) Heap wie in der Theorie auf S. 8 in einen Min-Heap transformiert werden.